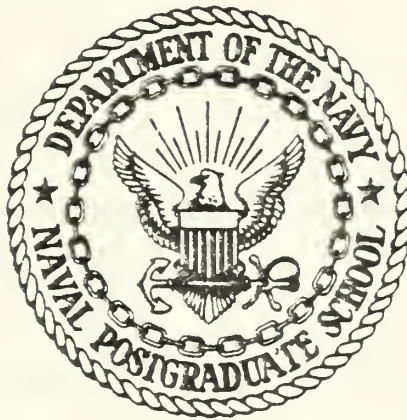




NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

SIMULATION OF DYNAMIC TACTICAL ROUTE
SELECTION WITH APPLICATION IN THE STAR MODEL

by

James Stephen Kramer

March 1979

Thesis Advisor:

S. H. Parry

T187807

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Simulation of Dynamic Tactical Route Selection with Application in the STAR Model		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis: March 1979
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) James Stephen Kramer		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE March 1979
		13. NUMBER OF PAGES 105
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Combat Simulation; Dynamic Programming; Route Selection; Simulation; STAR; Tactical Movement		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This thesis presents a deterministic simulation model for the dynamic selection of offensive tactical movement routes. The factors which influence route selection are identified, and the performance objectives that are to be optimized are defined. Alternative modeling concepts are investigated, and one method is selected for implementation. The organization,		



data structure, and computational aspects that were developed to implement this concept are explained. A FORTRAN program listing of the route selection model is presented. The test situations in which the model was exercised are documented, and the conclusions resulting from these tests are presented. Potential expansions and applications of this dynamic route selection model are also discussed.



Approved for public release; distribution unlimited

Simulation of Dynamic Tactical Route Selection
With Application in the STAR Model

by

James Stephen Kramer
Captain, United States Army
B.A., Iowa State University, 1972

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1979

1817
-1

ABSTRACT

This thesis presents a deterministic simulation model for the dynamic selection of offensive tactical movement routes. The factors which influence route selection are identified, and the performance objectives that are to be optimized are defined. Alternative modeling concepts are investigated, and one method is selected for implementation. The organization, data structure, and computational aspects that were developed to implement this concept are explained. A FORTRAN program listing of the route selection model is presented. The test situations in which the model was exercised are documented, and the conclusions resulting from these tests are presented. Potential expansions and applications of this dynamic route selection model are also discussed.

TABLE OF CONTENTS

I.	INTRODUCTION-----	7
II.	THE ROUTE SELECTION PROCESS-----	10
III.	MODELING CONCEPTS-----	17
IV.	DESCRIPTION OF THE MODEL-----	30
	A. MODELING CONCEPT-----	32
	B. STRUCTURE OF THE MODEL-----	34
	C. IMPLEMENTATION OF THE MODEL-----	55
V.	DOCUMENTATION OF TEST RESULTS-----	61
VI.	CONCLUSIONS AND RECOMMENDATIONS-----	77
	APPENDIX A: FLOWCHART-----	80
	APPENDIX B: LIST OF VARIABLES-----	87
	APPENDIX C: COMPUTER PROGRAM-----	91
	LIST OF REFERENCES-----	102
	INITIAL DISTRIBUTION LIST-----	103

LIST OF FIGURES

1. DYN-TACS Route Selection Model-----	13
2. Flexible Route Concept-----	18
3. Fixed Alternate Route Concept-----	21
4. Sliding Pattern Concept (A)-----	24
5. Sliding Pattern Concept (B)-----	25
6. Information Flow-----	31
7. Route Selection Area-----	35
8. Route Selection Array-----	36
9. Assault Range-----	41
10. Neighbor Nodes-----	42
11. Shortest-Route Algorithm-----	45
12. Model Output-----	49
13. Route Re-evaluation-----	51
14. Update of CHANGE-----	53
15. Sample Terrain-----	64
16. Example Array-----	66
17. Least-time Route-----	67
18. Initial Route (A)-----	69
19. Initial Route (B)-----	70
20. First Route Re-evaluation-----	72
21. Second Route Re-evaluation-----	73

I. INTRODUCTION

Since World War II, the nature of warfare has changed significantly. Rapid technological developments have resulted in modern armies equipped with sophisticated, highly mobile weapon systems. The training and tactics that are necessary to employ these systems have also changed to keep pace with these developments. As a result, the modern battlefield is anticipated to be characterized by speed, violence, and rapid change. In this type of environment, the ability to move forces in response to the rapidly changing tactical situation is a critical factor in the ultimate outcome of the battle. On the small unit level, the ability to react quickly to an enemy force and to take advantage of the mobility that the unit is capable of is equally important to the success and survivability of that unit.

This increasing emphasis on mobility implicitly places a requirement on the combat simulations that are currently being used to validate tactics and weapon system characteristics. This requirement is that these simulations be able to realistically represent the effects resulting from flexibility in tactical movement. Yet, with one exception, all of the current mid and high resolution combat simulations employ some form of fixed, pre-determined pattern of movement. This thesis presents a model that is capable of dynamically selecting attack routes in response to changes in the tactical

situation. It is hoped that this model will prove useful in analyzing the relationships between flexible movement tactics and weapon system characteristics.

Chapter II briefly presents a description of the concept of route selection. The factors which influence route selection are identified, and the performance objectives which are to be optimized are defined. A measure of effectiveness that can be used to evaluate alternate routes is also presented. One method of quantifying these subjective concepts is described through the presentation of a summary of the DYN-TACS route selection model.

Chapter III outlines several modeling concepts that could be used to represent dynamic route selection. A general description of the procedures that would be required to implement each of these concepts is also presented. The advantages and disadvantages of each procedure are discussed, and, based on these characteristics, one concept is selected for further development.

The details of the selected modeling concept are developed in Chapter IV. The basic assumptions which underlie the chosen method of application are explained. In addition, the degree of interface that is required between the model and a parent combat simulation are discussed. This is followed by a description of the organization, data structure, and computational aspects of the proposed route selection model. Possible methods of implementation and their impact on the capabilities of the model are also discussed.

Chapter V documents the results of several test situations which were used to verify the FORTRAN program which has been developed from the route selection model. A brief description is provided of the interface that was required between the model and the STAR combat simulation which was used to exercise the model. The specific terrain and tactical situations that were provided as input to the model are also discussed. The optimum routes which were generated by the model are presented graphically, and the factors which influenced these routes are explained.

In Chapter VI, the conclusions drawn from the initial tests of the route selection model are discussed in terms of their implications for the future use and expansion of the model. Potential areas for future development of the model and areas requiring additional analysis are also described.

A flow chart of the route selection process is presented in Appendix A. Appendix B lists and defines the major variables that are used in the computer program of the model. A source listing of this program is provided in Appendix C.

II. THE ROUTE SELECTION PROCESS

The development of a computer simulation to model the route selection process must generally follow three basic, yet interrelated, steps:

1. Definition of the factors that influence route selection.
2. Quantification of these factors and their functional relationships.
3. Development of a procedure to use these quantifications to generate a realistic representation of a route of advance.

Although the objective of this thesis is to analyze methods that can be used in the last step of this sequence, it is first necessary to address steps one and two. This chapter will present the basic elements required by these two steps. The subsequent chapter will be devoted to an analysis of alternative procedures that can be used to model the selection of a route of advance.

The objectives of this chapter will be met by presenting a summary description of the route selection subroutine that has been developed for the DYN TACS combat simulation model. DYN TACS is the only combat model that currently has the capability of dynamically selecting movement routes in response to changes in the tactical situation [Ref. 1]. Thus, this method of presentation not only provides an appropriate framework within which the variables and functional relationships can be presented, but it also provides a summary

of previous research in the simulation of the route selection process. The description of the DYN-TACS route selection model used in this chapter is taken from the final study report presented in Ref. 2.

In the DYN-TACS route selection model, the basic concept being modeled is that the attacking maneuver unit seeks to move as rapidly as possible and with minimum exposure to enemy weapons, until it is within effective range of the objective. This concept defines the two primary measures of performance: time and exposure. These two performance objectives are related. The selection of a route of advance involves making a trade-off between the time it will take to traverse a specific route and the degree of exposure to enemy weapon systems along that route. This relationship is used to define a measure of effectiveness called tactical difficulty. The tactical difficulty is expressed as the product of two terms:

$$DIJMN = T (1 + E) \quad (2-1)$$

where $DIJMN$ = relative tactical difficulty to travel through a coordinate system from point (I,J) to point (M,N) ,

T = estimated travel time between (I,J) and (M,N) , and

E = difficulty resulting from exposure to enemy weapon systems in moving from (I,J) to (M,N) .

The goal then is to select the route which produces the minimum tactical difficulty. Notice that if there is no enemy influence (in which case E is equal to zero), the optimum route is the route with the least travel time. Also, the tactical

difficulty is a measure of travel time plus the product of time and exposure. Thus, time is the primary performance objective that is minimized.

The two performance objectives used in this model are defined in terms of those facts that a unit commander would realistically have available to him. For example, the exposure factor is based on those enemy weapon systems that he has actual knowledge of, and on those specific locations that have previously been identified as suspected enemy positions. The travel time, which is a function of terrain and mobility factors, is estimated by considering only the dominant terrain features. Specific features become relevant only as the commander becomes aware of them.

The actual route selection process begins by defining a feasible region, called the route selection area, from which the optimum route will be identified. This feasible region is restricted to an area on either side of a fixed axis of advance. The route selection area is identified by an array of points as shown in Figure 1. The center column of the array is aligned with the grid point representing current position and with a point on the axis of advance which is a fixed (input) distance from the current position. This fixed distance represents the planning horizon for the commander's decision process.

Within this framework, a route is defined as a sequence of points in the array which leads from the current position to any point in the last row of the array. Of course, if the objective is within the route selection area, the route

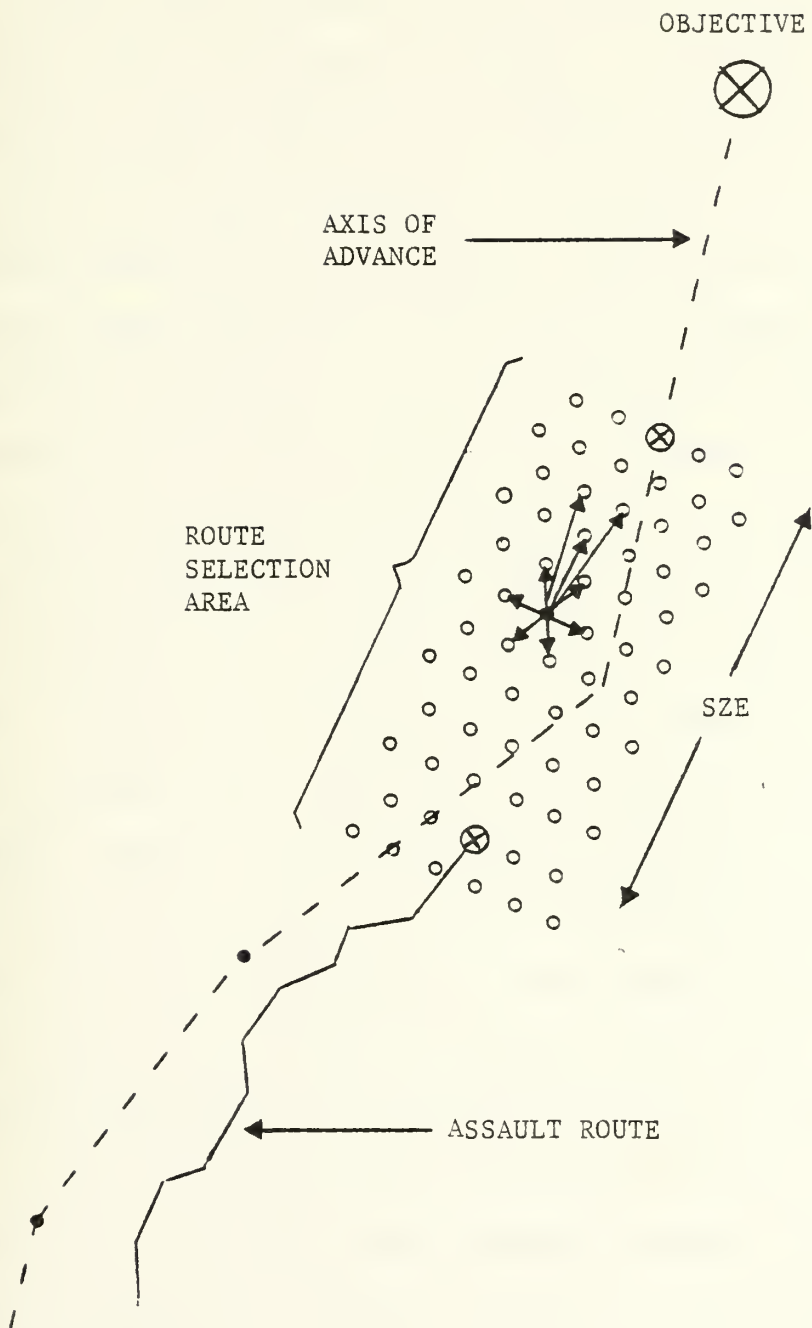


Figure 1. DYN TACS Route Selection Model

must terminate at the objective. The identification of a feasible route is based on the concept that from any given point in the array there are nine possible neighboring points to which a unit is allowed to move. An example of these permissible route segments is shown for one such point in the array depicted in Figure 1. From the large number of possible combinations of segments, the one route that produces the least tactical difficulty is selected. A dynamic programming algorithm is used to efficiently find this route. A description of this algorithm is contained in Ref. 2.

The tactical difficulty associated with any potential route is the sum of the tactical difficulties for the individual segments that connect the points which define that route. If the objective is not within the route selection area, an additional time factor is added to this sum to represent travel time beyond the last point in the route. Equation 2-1 is used to calculate the individual difficulty values. The time component in this equation is the estimated travel time for the unit to traverse the route segment. The exposure component is calculated at the end point of the segment. It is the sum of the difficulty factors resulting from exposure to individual enemy weapon systems. Each weapon system is assigned a weight relative to its effectiveness. A weapon contributes its weight factor to the exposure at a point if it is both within effective range and also inter-visible with that point. If these conditions are not met, no weight is added by that particular weapon.



It is important to note here that when an attacking unit comes within assault range of the objective, it no longer seeks to avoid exposure to the enemy. The unit now seeks fields of fire so that it may engage the enemy. To represent this tactic, DYN-TACS assigns negative exposure weights to the weapon systems located at the objective. This procedure results in an optimum route that tends to avoid enemy contact while the unit is moving toward the objective yet tends to seek maximum contact when the objective is within final assault range.

Once this least difficult route has been selected from within the route selection area, it is used to guide the actual movement of the attacking unit. The route is re-evaluated, and typically extended, whenever specific criteria are met. For example, a new route is generated before the unit traverses the entire route selection area. The distance that the unit is allowed to travel before this occurs is specified as an input parameter. A new route is also selected whenever the unit encounters an obstacle or whenever the number of detected or destroyed weapon systems reaches a specified threshold level. As these criteria are successively met, optimum route increments are generated. This procedure is continued until one such route terminates at the objective.

The route that is generated by this procedure is optimized in terms of two specific performance objectives. This route is a reflection of these objectives and the assumptions



that underlie the relationships and procedures that are used in this model. In general, the description of the route selection philosophy that has been presented in this chapter is intuitive to military judgement. This is not to say that the concepts used in the DYNTACS model are assumed to be the most realistic representations of the decision process which actually occurs when a commander selects an attack route. However, improvement or verification of these ideas will be left as a topic for future study. The concepts that have been presented here will be used, with minor modification, as a basis for comparative analysis in the subsequent chapters of this thesis.

III. MODELING CONCEPTS

With the background material from the previous chapter, it is now possible to begin to explore some of the various approaches that can be used to model the decision process involved in selecting a route of advance. In this chapter, four general concepts will be presented. The order in which these concepts appear represents a learning process. Successive models are attempts to achieve a better representation of the route selection process and to structure the problem in a manner more appropriate to the solution techniques that are available.

Most mid and high resolution combat simulations being used by the U.S. Army utilize a fixed, input route to control movement [Ref. 1]. This procedure can be expanded to allow this route to be shifted in response to changes in the tactical situation. Figure 2 shows how this first concept might be depicted. In the figure, the primary route of advance has been altered to reflect the influence exerted on that route by three enemy weapon systems. The new route that has been selected represents an attempt to reduce the tactical difficulty involved in moving to the objective. It reflects the trade-off between exposure and travel time.

This new route is developed by evaluating the difficulty due to exposure to enemy influence factors at discrete points spaced equally along the original route. These influence

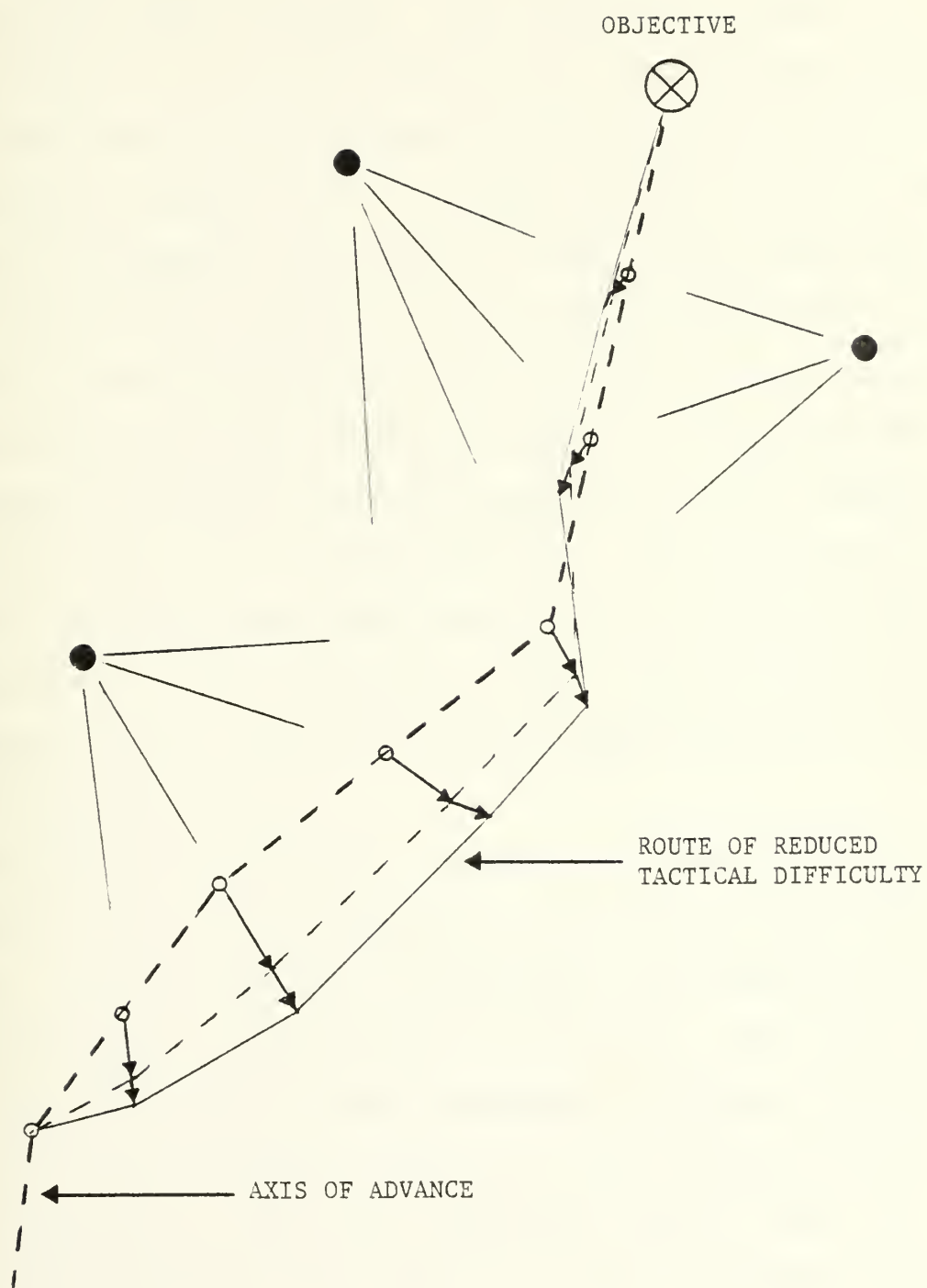


Figure 2. Flexible Route Concept



factors might be weapon systems, minefields, or obstacles. At each point, a vector is used to represent each such known or suspected enemy influence. The length of a vector is proportional to the weight assigned to the respective influence factor and to its distance from the point on the route. The direction of the vector is simply the direction from the center of influence to the point. These vectors form the components of one resultant vector which determines the direction and relative distance that the route is to be moved at the particular point being evaluated. A predetermined increment of distance is used to specify the total change allowed in the route at each iteration. This distance increment is divided among the points on the route in proportion to the length of the resultant vectors. These vectors and distances define a new set of points which are used to calculate the new route.

The tactical difficulty value due to exposure and time along this new route is compared to the value of the previous route. If the difficulty has decreased, the procedure is repeated until the value begins to increase. When this occurs, the allowable distance increment is reduced, and the entire process is repeated until the final change in tactical difficulty is less than a specified level. The last route that has been identified is a route of minimum tactical difficulty.

This flexible route concept will select a route in the immediate vicinity of the primary axis of advance. In terms

of optimization theory, this route may be only a local optimum. The desirability of this route is highly dependent on the location of the initial route and on the specific tactical situation. For example, if a route initially passes between two strong factors of roughly equal influence, it will always remain between these factors. The final location of the route will be at the point where the two component vectors counteract each other. Also, a resultant vector allows the route to be moved in only one direction. This could produce a long, meandering route if the sources of influence were staggered along the route of advance. Thus, this model does not necessarily evaluate routes that could circumvent the sources of influence and possibly reduce the tactical difficulty.

In order to provide a wider range of potential routes, an alternate method of modeling dynamic route selection can be used. Under this second concept, a network of routes is identified and provided as input data to the model. Figure 3 provides an example of such a set of routes. The number and shape of these routes, or route segments, is chosen to reflect the specific terrain features and tactical situations that might be expected to be encountered. The exact pattern of routes that is used could be designed to fit the needs of the specific application of the model. The advantages of such a procedure, however, should be weighed against the time and resources devoted to planning and identifying these routes. A more practical approach might be to design a set

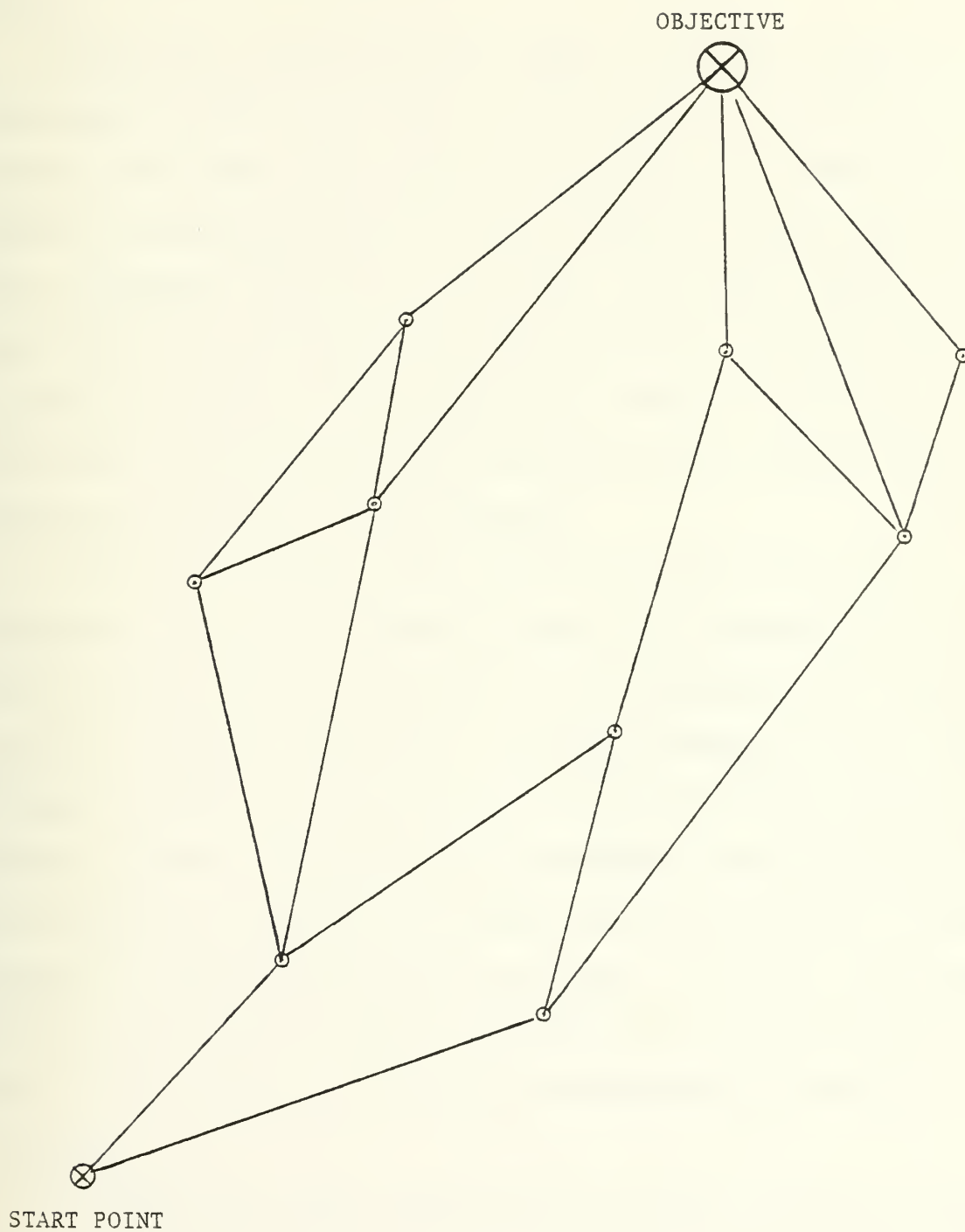


Figure 3. Fixed Alternate Route Concept



of fixed patterns. The requirements of the specific application would then determine which pattern would be most appropriate.

Regardless of the procedure used to select the set of alternate routes, the method of identifying the optimum route remains the same. The pattern is treated as a network in which the intersection points are defined as the nodes and the connecting segments as the arcs. The arc "lengths" are the respective tactical difficulty values. A dynamic programming formulation is an efficient method of finding the "shortest" route connecting the nodes which represent the current position and the objective.

Under this concept, the alternate routes are fixed. Therefore, a planned movement route can be changed only at one of the intersection points. The smoothness and flexibility of the optimum route is highly dependent on the programmer's judicious placement of the intersection points and route segments. This potential weakness could be minimized by developing a network of routes which consists of a large number of evenly spaced arcs and nodes. However, this would require substantial effort to define the network and to convert it to a form that could be programmed and stored in a computer.

In order to reduce resource requirements, and yet obtain detailed route representation, a moving pattern concept can be employed. In this third model, the fixed routes are replaced by a set of parameters which defines a fixed pattern.

This pattern slides along the selected route as the tactical unit advances. When the route is re-evaluated, the parameters are used to define specific alternate routes relative to the current position of the unit. In contrast with the previous model, these routes are contained within an area of reduced size. Thus, an equal degree of detail can be obtained using a network with fewer nodes and arcs. Figure 4 shows an example of this concept. The pattern represented in the figure is designed to always terminate at the objective. As the distance between the unit and the objective decreases, the pattern is compressed. This provides a more detailed representation of the route as the objective is approached and as the intensity of combat would reasonably be expected to increase. However, this approach provides a relatively coarse pattern of routes in the initial stages of movement.

An alternative procedure is to use a pattern of fixed length. This provides a constant degree of detail as the pattern slides along the route. Figure 5 provides an example of an open-ended pattern of fixed size. A shortened route section is selected from the alternatives available within the pattern. This approach is particularly appropriate if the concept of a limited planning horizon is assumed. Under this concept, the size of the pattern is fixed to reflect the maximum area that a commander would reasonably consider when selecting his route of advance. Successive applications of this concept will extend the route by continually adding route increments. Since the end of the pattern will

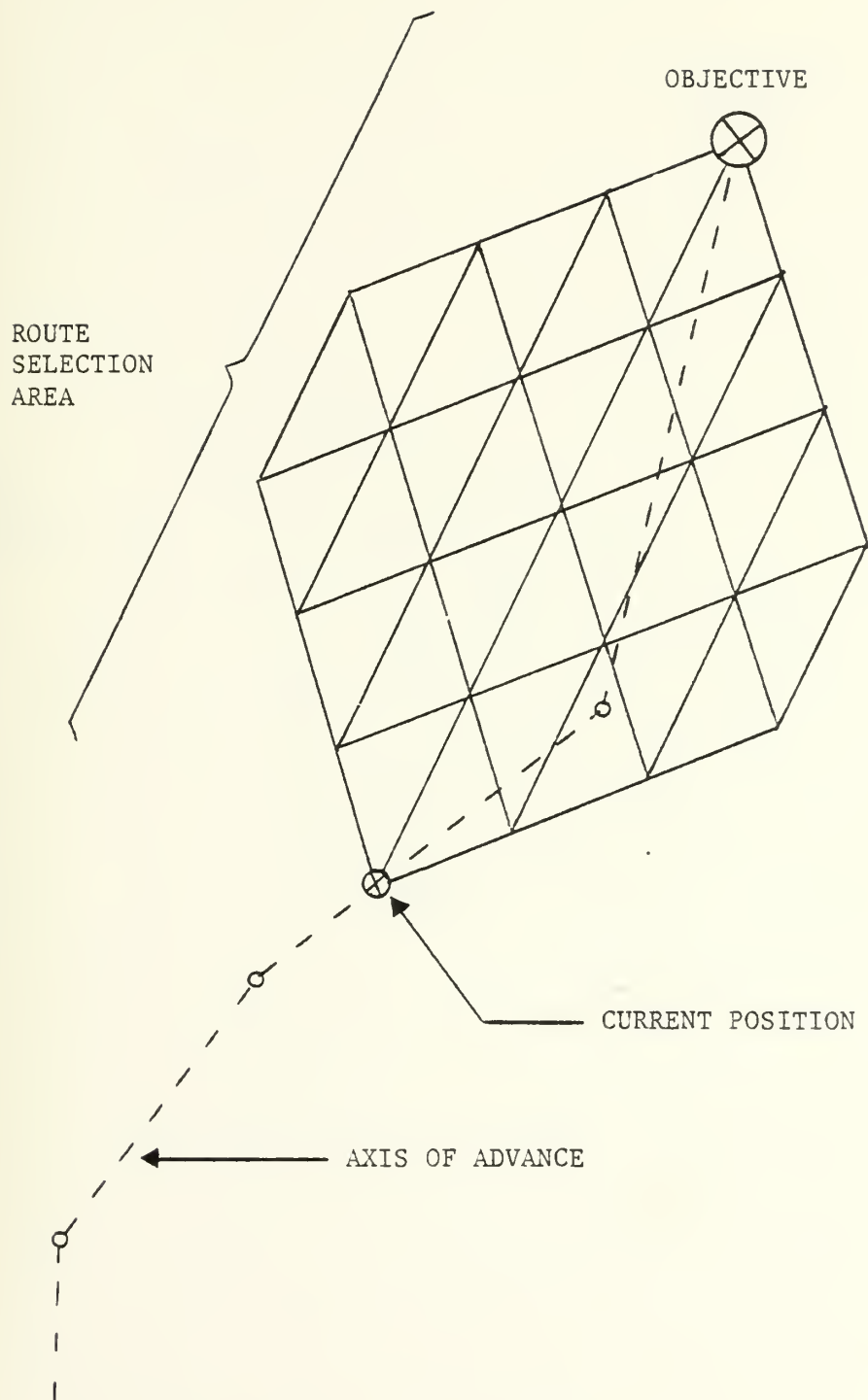


Figure 4. Sliding Pattern Concept (A)

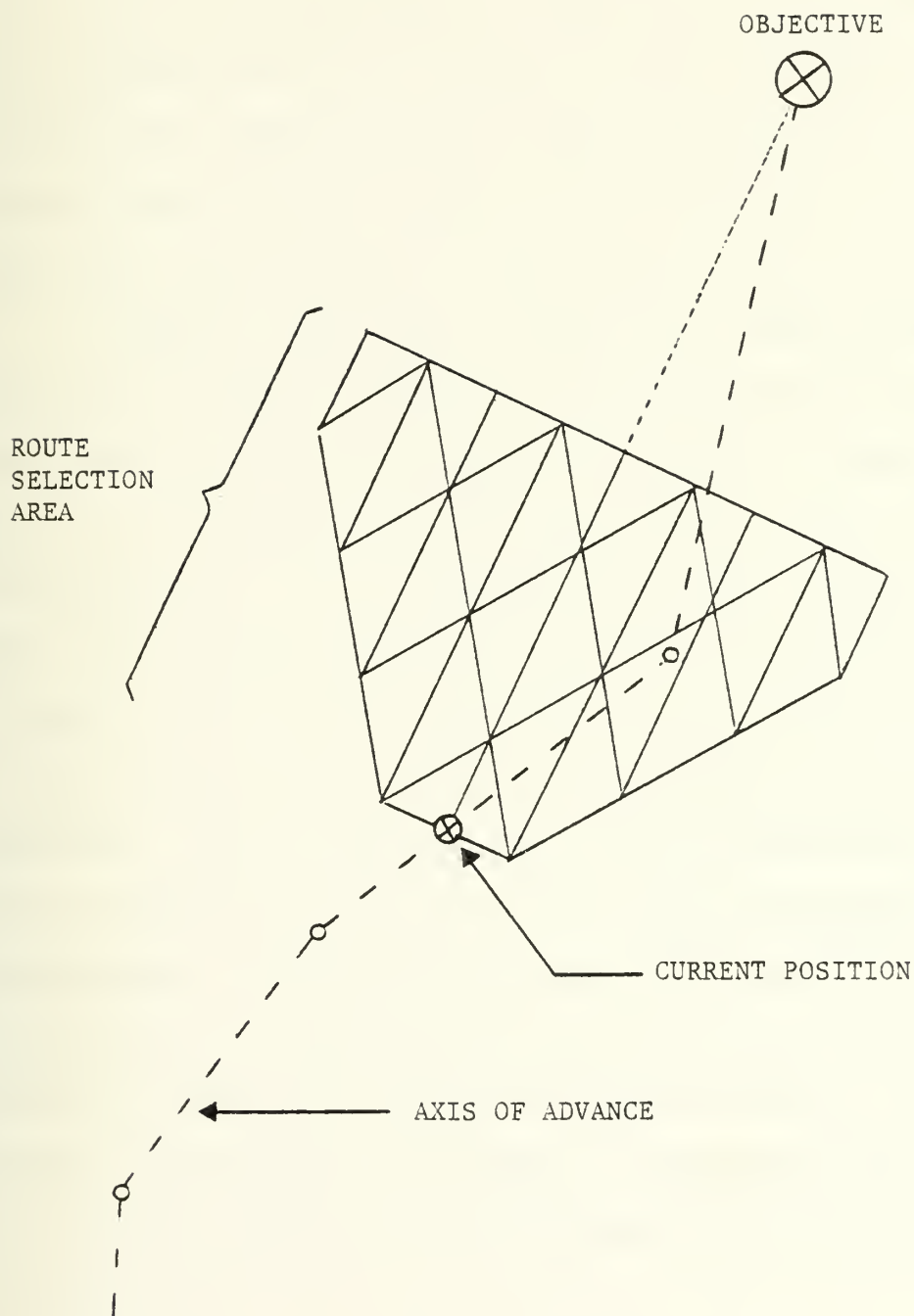


Figure 5. Sliding Pattern Concept (B)



typically fall short of the objective, a method must be selected to orient the routes in an appropriate direction. This can be accomplished by centering the end of the pattern on a prescribed axis of advance or on a straight line connecting the current position with the objective. On the final iteration, the pattern must be modified to insure that the last route increment terminates at the objective.

In general, the basic concept of a sliding pattern of alternate routes is merely an extension of the previous model. However, this concept does provide a framework which can allow a greater degree of flexibility and detail in the optimum route. It also allows a smoother transition to a new route, since the initial point of the pattern is always located at the point representing the current position. As in the previous model, a set of alternate patterns can be employed, and the optimum route can be efficiently identified through the use of shortest-route algorithms as solution techniques. However, both models rely essentially on the same method of representing feasible routes. Networks composed of explicitly defined arcs and nodes are used to describe and evaluate the routes. A more efficient structure is available, and it forms the basis for the next modeling concept.

In the fourth, and last, approach to the route selection problem, the network of feasible routes is defined by using an array of grid points to represent the nodes and a decision rule to identify permissible arcs. The same concept is used in the DYN-TACS route selection model. Figure 1 shows the

specific method used in that model. This method is a discrete grid approach applied to the sliding pattern concept. However, this same structure can be used in a wide range of alternative formulations. For example, the fixed route and the compressible, sliding pattern concepts can also be expressed in this type of framework. In addition, a great deal of flexibility is allowed within any specific concept through the selection of the size of the array, the spacing of the grid points, and the decision rule used to identify allowable neighboring points. Since this concept simply employs an alternate form to represent the network of routes, dynamic programming techniques can still be used to efficiently generate the optimum route.

Since the discrete grid approach can be applied in either of the previous two models, it can be used to reflect a wide variety of route selection philosophies. The primary advantage afforded by this approach is efficient computer storage. This is due to the special structure resulting from the uniform array of grid points and the uniform pattern of route segments.

All four of the concepts that have been presented in this chapter have dealt with piece-wise linear routes and discrete point evaluations. An alternative approach is available, and merits at least a brief discussion in order to complete the analysis of potential alternatives.

A continuous representation of the route selection process can be described in terms of the calculus of variations.

In this formulation the objective is to find a continuous function to represent the route of advance. The optimum route is that route which minimizes the integration of a continuous tactical difficulty function over that route. Assuming, for the time being, that a continuous difficulty function is available, the solution to this particular formulation requires the solution of second-order, non-linear, differential equations. Except in very special cases, a closed form solution to these equations does not exist [Ref. 3]. Discrete approximations must be used to achieve a numerical solution. References 2 and 3 offer various discrete solution techniques which might apply to this route selection problem. However, these techniques would require a continuous difficulty function.

Any effort to develop a continuous tactical difficulty function implicitly assumes that such a function would provide an improved method of either representation or computation. The difficulty associated with representing intervisibility in a continuous manner eliminates the possibility of this concept providing a more accurate representation. This fact counteracts any advantage that might result from improved computational efficiency, if such an improvement is in fact possible. Thus, the formulation of the route selection problem in a continuous framework does not produce the advantages that might seem apparent at first glance.

Because of the efficiency and flexibility afforded by the discrete grid formulation, this structure will be used

in the following chapter where a program to model the route selection process will be developed.

IV. DESCRIPTION OF THE MODEL

In this chapter, a model of the route selection process will be presented. This model is designed to complement a larger combat simulation model by selecting optimum movement routes in response to the tactical situations which are represented in the simulation. This process requires a moderate degree of interface between the model and the simulation. The relationship and information flow between the route selection model and the various components of the simulation are depicted in Figure 6. The model draws on terrain, intelligence, and movement information which is stored or generated within the simulation. This data is used to evaluate potential movement routes. The model then provides a description of the optimum route to the simulation. For the purpose of this thesis, rather than duplicate the simulation functions and data sets, it has been assumed that the necessary information is available to the route selection model. Moreover, wherever the model requires a specific format for this data, it has been assumed that the information is either currently available in proper format or can be modified to meet this requirement.

The description of the route selection model is presented in three parts. In the first section of this chapter, the general modeling concepts and procedures are briefly presented. The subsequent section presents a detailed description of

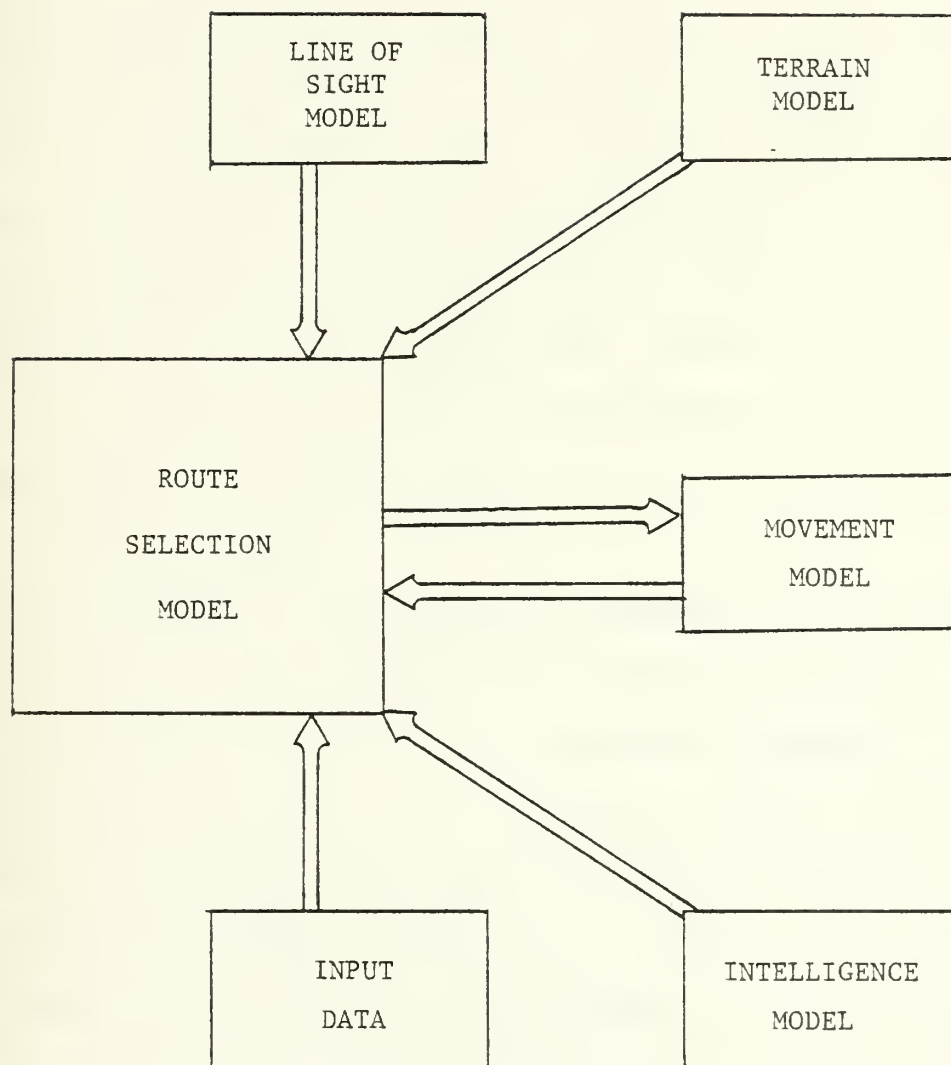


Figure 6. Information Flow



the structure of the model. In the last section, methods of implementing the route selection model are discussed.

A. MODELING CONCEPT

The tactical situation being modeled is that of a single maneuver element moving through an assigned sector from a start point to an objective. The basic assumption in this model, as in the DYN-TACS model, is that a maneuver element seeks to move to its assigned objective along that route which minimizes the travel time and the exposure to enemy influences. It is also assumed that tactical difficulty, which is a function of these two performance objectives, is a good measure of effectiveness by which potential movement routes can be compared.

This concept is applied to the scenario being modeled through the use of a discrete grid representation of the feasible route selection area. Since the maneuver element is assigned a sector of the battlefield within which it is allowed to maneuver, this sector defines the feasible region from which the element must select a route. This fact is reflected in the route selection model by establishing an array of grid points which cover the entire maneuver area. This array remains constant for the total period of time that the maneuver element is moving from its starting point to its objective.

From within this array, an initial optimum route is selected which leads to the objective. This route is chosen to minimize the cumulative tactical difficulty that the

maneuver element will encounter. This difficulty is a reflection of the travel time and known or suspected enemy sources of exposure. The functional form chosen to represent the tactical difficulty is the DYN-TACS relationship:

$$TD = T(1 + E)$$

where TD = tactical difficulty,

T = travel time, and

E = relative exposure weight.

However, for that portion of the route selection array which is within assault range of the objective, the exposure values are set to zero. Thus, within this area, the route is optimized only in terms of travel time. This concept is based on the assumption that when the maneuver element reaches a specified assault distance from the objective, it seeks to occupy that objective as forcefully and quickly as possible. Thus, the route which is generated by this model will tend to minimize time and contact with enemy elements while moving toward the objective, and will then tend to close on the objective as quickly as possible.

The initial route that has been selected in this manner will be periodically re-evaluated in order to reflect any changes in the tactical situation that may have occurred. Whenever this is done, the route selection array that has already been established is re-used. The maneuver element's position at the time the re-evaluation takes place is used as the new start point for a route to the objective. Thus, at successive re-evaluations, the portion of the array which is actually utilized is successively reduced. This



can be seen for the array shown in Figure 7. This concept eliminates the need to generate a new array at each iteration of the route selection process. In addition, the travel times and exposure values which have already been computed do not have to be totally recalculated. The existing values are merely updated to reflect the changes that have occurred since the last route was selected. Also, this needs to be done only within the reduced route selection area. This route re-evaluation process is repeated as necessary until the maneuver element is sufficiently close to the objective.

B. STRUCTURE OF THE MODEL

The description of the model contained in this section follows the general sequence of events that occur as a route is selected and then successively re-evaluated. A flow chart is presented in Appendix A which depicts this flow of control.

1. Route Selection Array

The first step in the route selection process is to identify the initial route selection area. This requires that input values be assigned to the variables XO , YO , XT , YT , $SECTOR$, $RSPACE$, and $CSPACE$. Figure 8 shows how these values are used to construct the route selection array. The number of rows that are contained in the array is determined by the desired row spacing, $RSPACE$, and the distance between the initial point (XO,YO) to the terminal point (XT,YT) . In a similar manner, the number of columns is determined by the desired column spacing, $CSPACE$, and the width of the sector



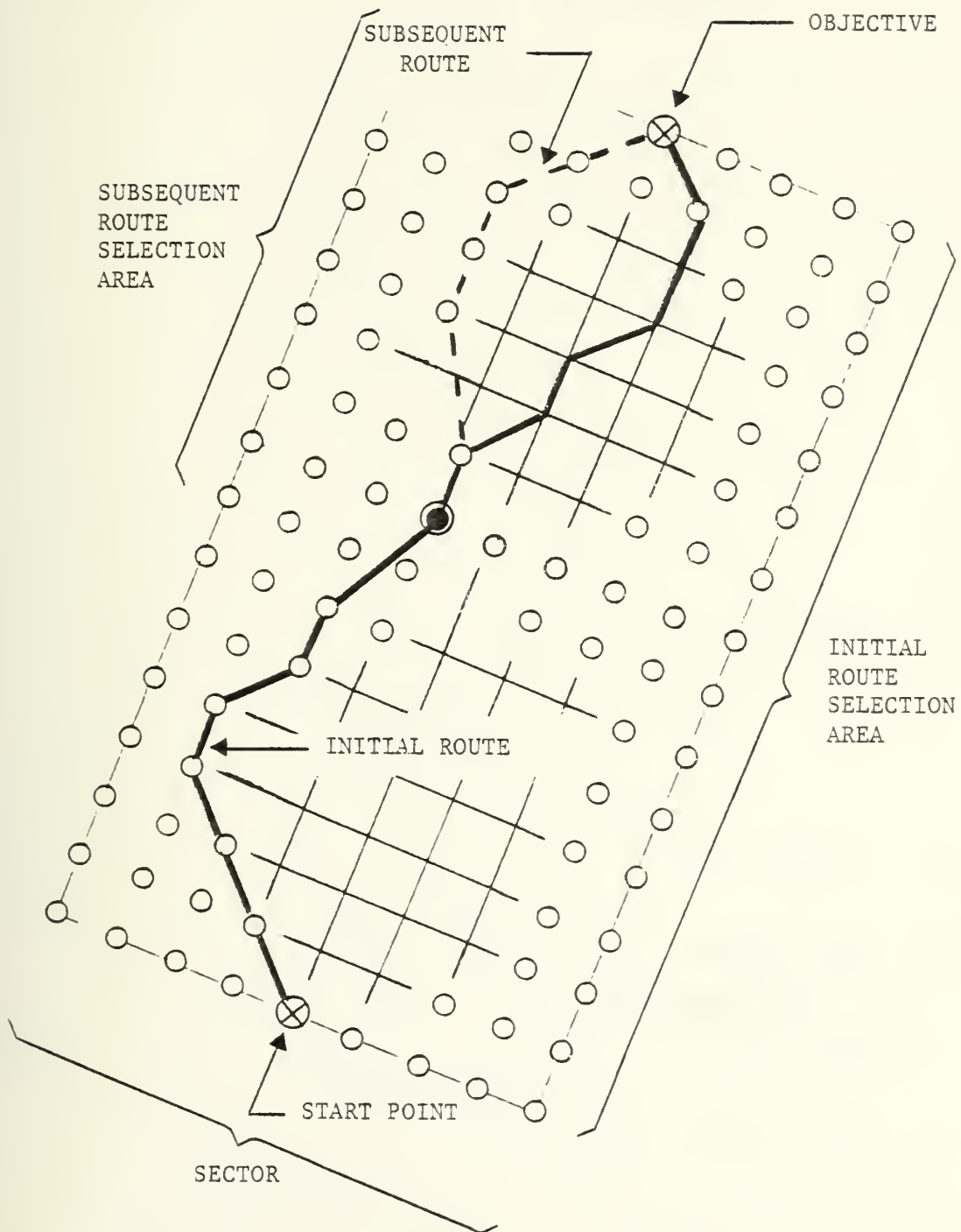


Figure 7. Route Selection Area



$$NROW = \text{ROUND} \left[\frac{D}{RSPACE} \right] + 1$$

$$NCOL = \text{ROUND ODD} \left[\frac{\text{SECTOR}}{CSPACE} \right] + 1$$

$$RFIX = \frac{D}{NROW - 1}$$

$$CFIX = \frac{\text{SECTOR}}{NCOL - 1}$$

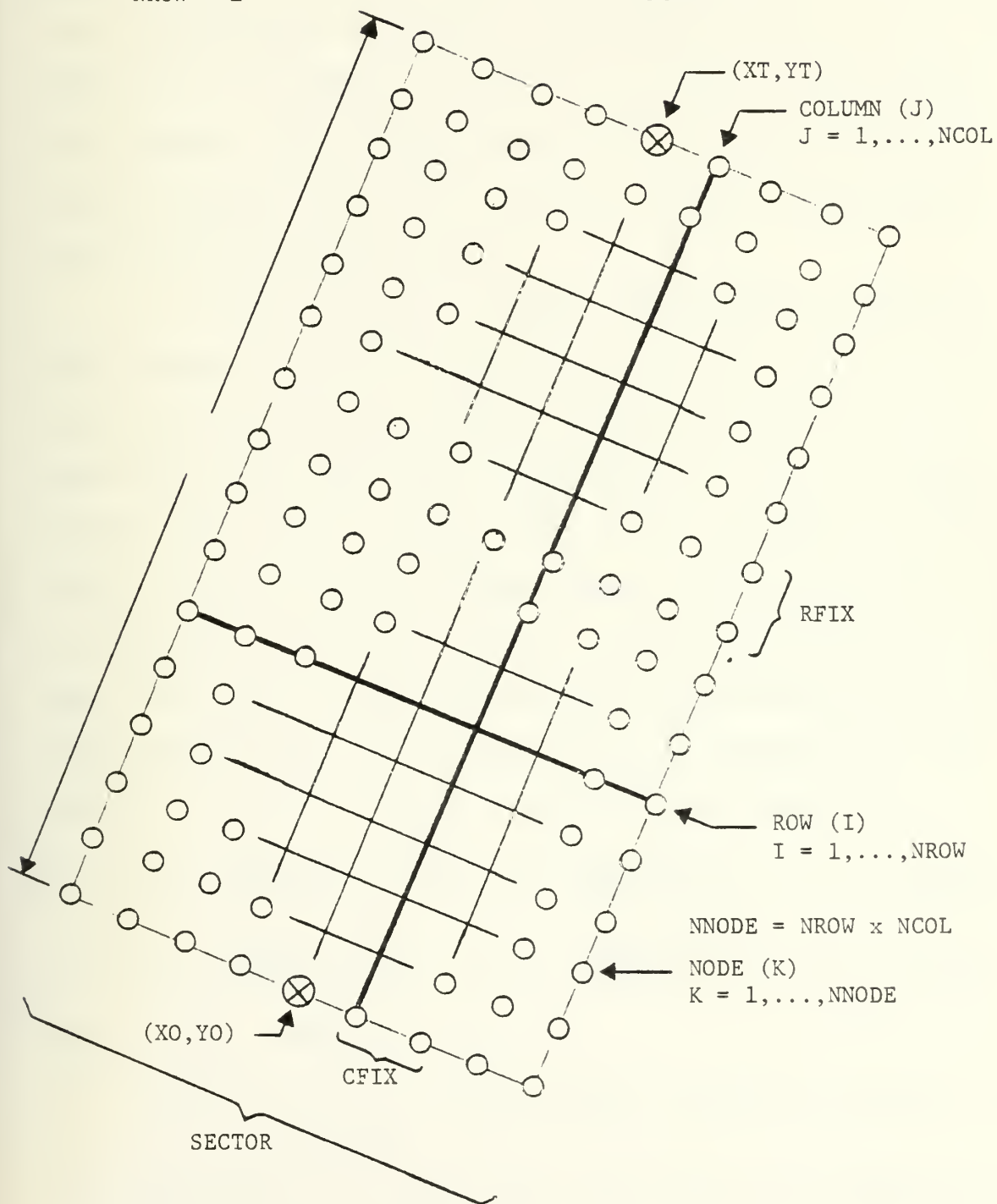


Figure 8. Route Selection Array

which is stored in the variable called SECTOR. Since neither the width nor the length of the sector will typically be an integer multiple of the spacing variables, the actual spacings are adjusted to achieve the nearest integer values. In addition, the number of columns is rounded to the nearest odd integer. This insures that the initial and objective points coincide with the center grid points in the first and last rows respectively.

At this point, it is also advantageous to assign node numbers to the grid points in the array. This is accomplished by numbering the points sequentially by rows. Associated with each node in the array will be an exposure weight and a set of travel times to its allowable neighboring points. But before these values can be determined, it is necessary to convert the row and column coordinates of each grid point to the corresponding coordinates in the X-Y coordinate system. These values can be computed for any point (I,J) in the array by the following equations:

$$X = X_0 + \frac{I - 1}{NROW - 1} (X_T - X_0) + \frac{J - J_{HALF}}{NROW - 1} (Y_T - Y_0) (RATIO)$$

$$Y = Y_0 + \frac{I - 1}{NROW - 1} (Y_T - Y_0) - \frac{J - J_{HALF}}{NROW - 1} (X_T - X_0) (RATIO)$$

where $J_{HALF} = \frac{NCOL + 1}{2}$, and

RATIO = ratio of actual column spacing to actual row spacing

The coordinates for each node in the array are computed and stored for future reference.



2. Exposure

In order to compute the exposure weights for each node, two data lists must be available. These lists reflect the two sources of exposure which are considered in the model. A list called SUSP contains the coordinates of a set of pre-determined, suspected enemy positions. These positions might represent terrain features which could be advantageous to the enemy and therefore should be avoided by the maneuver element. This list is provided as input to the model. The second list, called STATUS, reflects the maneuver element's concept of the current tactical situation. It should be noted that the element's current perception of the battlefield should include not only influences that are currently visible to the element, but also those influences which have been detected previously and those which the element has been informed of through its intelligence network. STATUS contains the locations and types of these enemy influences. These influences might be weapon systems, obstacles, or minefields. The information which is stored in this list is continually changing as combat is simulated in the parent model. However, at the time of the first route selection, this list is fixed to represent the initial tactical situation. At this initial stage, STATUS may very well be empty. Therefore, the list is scanned to determine whether it needs to be considered in the exposure calculations.

The actual computation of the exposure factors must be based on the subjective selection of a relative weight

function. In addition, the relationship between weapon effectiveness and intervisibility must be specified. To avoid restricting the model to one exposure philosophy, the route selection model requires that a user-prepared subroutine be provided for computing exposure weights. This subroutine, called EXPOSE, must be designed to receive input data consisting of the coordinates of the node being considered, the coordinates of the enemy influence, and an identification of the type of that influence. With this information, the subroutine must be capable of calculating and returning an exposure weight. Within this general framework, the programmer may specify the functional relationships that seem most appropriate. Typically this subroutine will utilize line of sight information from the terrain model in the simulation. It may also use a set of weapon effectiveness equations which compute exposure weight, as a function of range, for those systems represented in the simulation. A wide range of alternative structures and methods of representation can be implemented in the subroutine.

The subroutine EXPOSE is called to provide exposure weights for the nodes in the route selection array. At any node, the subroutine is called to compute an exposure factor for each item contained in the list SUSP and, if necessary, the list STATUS. The sum of these factors is the exposure weight which is assigned to that node. However, not all nodes require this computation. For those points within assault range of the objective, an exposure value of zero



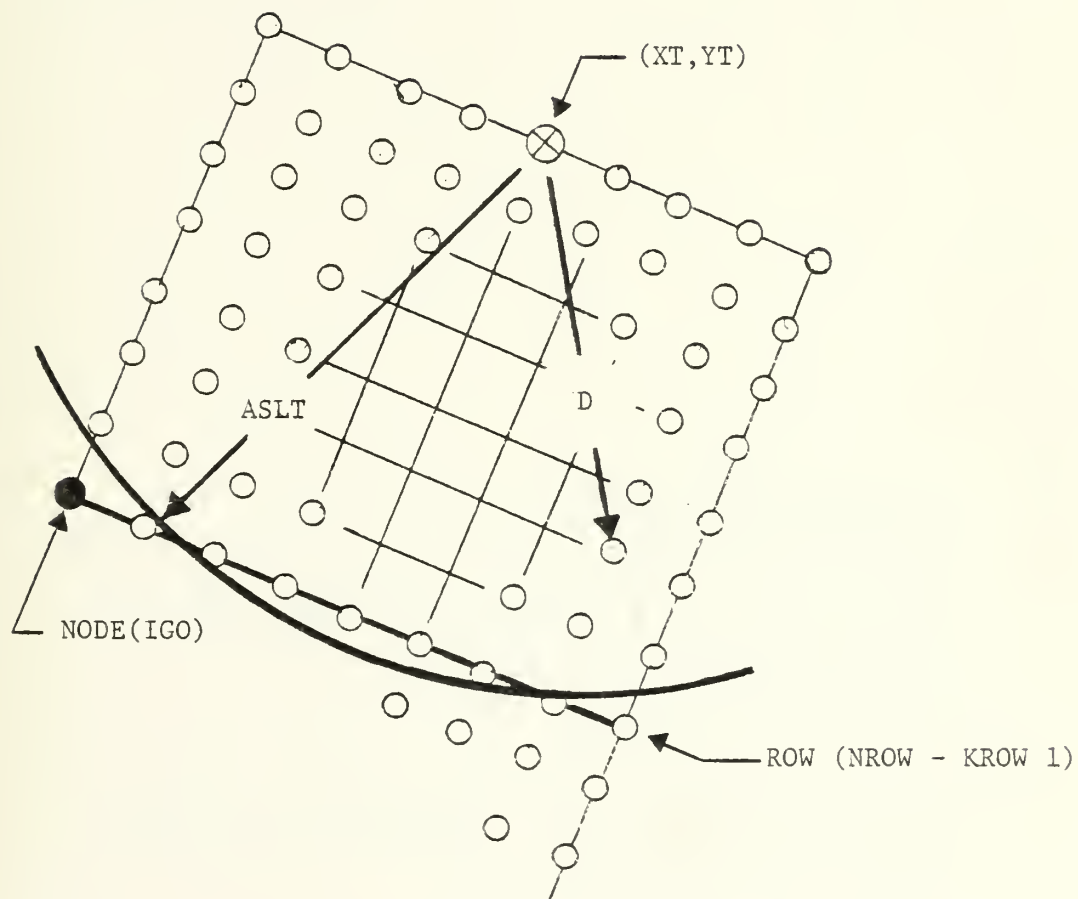
is assigned. An input variable called ASLT is used to store the specific assault range to be employed at the objective. This variable can be used to identify and flag those nodes for which exposure calculations can be skipped. Figure 9 shows how this procedure is applied. After the appropriate nodes have been flagged, exposure weights can be assigned to every node in the route selection array. At the conclusion of this process, the contents of STATUS must be copied into a similar list called CHANGE. This serves to record any temporary influences which have been considered in this iteration of the route selection process. The CHANGE list will be required when the initial route is re-evaluated.

3. Travel Time

In order to compute the travel times which are required in the tactical difficulty equation, it is first necessary to define the pattern of feasible arcs that are to be allowed at each node. The route selection model uses a subroutine called NABOR to identify the appropriate set of neighboring nodes which can be reached from any given node. The subroutine accepts a node number as an input argument and returns a list called NBR which contains the node numbers of its allowable neighbors. Within the subroutine, an array called ARC is used to identify these points. This array contains the relative row and column locations for the "neighborhood". Figure 10 provides one example of how the ARC array might be structured and of how the data are utilized to identify the appropriate node numbers. If the

$$KROW = \text{INTEGER} \left[\frac{ASLT}{RFX} \right] + 1$$

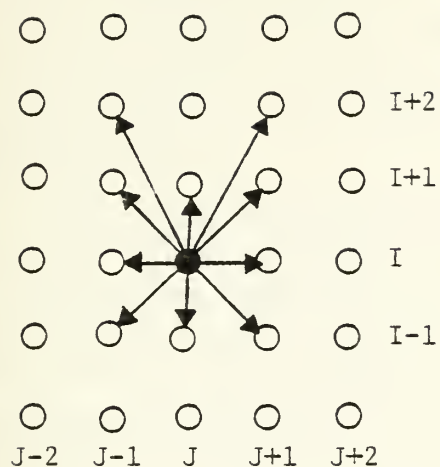
$$IGO = (NROW - KROW) NCOL + 1$$



IF $D(I) \leq ASLT$
 SET FLAG FOR NODE(I)
 I = IGO, ..., NNODE

Figure 9. Assault Range





INPUT: NNBR = 9

NBR(K); K = 1, ..., NNBR

N = NODE + ARC(1,L)

L = 1, 2, 3

NBR(K) = N + (NCOLS x ARC(L,M))

M = 3, ..., (ARC(2,L) + 2)

ARC				
RELATIVE COLUMN POSITION	←	-1	0	-1
# NODES THIS COLUMN	←	4	1	4
RELATIVE ROW POSITION	{	-1	1	-1
		0		0
		1		1
		2		2

Figure 10. Neighbor Nodes

pattern of potential route segments includes some points which lie outside of the route selection array, the subroutine will place a flag value in the appropriate locations in the NBR list.

The actual values for the estimated travel times to the neighboring nodes are obtained from the movement model which is part of the parent model. This movement model must be structured so that it can accept, as input arguments, the coordinates of the current node and one of its neighbor nodes. A travel time must then be computed based on the terrain between these points and on the accuracy desired in the estimation process. A set of estimated times is calculated and stored for every node in the route selection array.

4. Route Selection

With the time and exposure data, it is now possible to begin the evaluation of potential routes. From the many possible route combinations, the optimum route can be efficiently identified through the use of a shortest-route algorithm from network theory. For networks with non-negative arc weights, in this case non-negative tactical difficulties, Dijkstra's algorithm is the most efficient method available [Ref. 3,5,6]. This algorithm uses a label setting procedure to permanently label those nodes to which an optimum route has been determined. At each iteration, one additional node is so identified. The procedure is repeatedly applied to the network until the objective node has been permanently labeled, indicating that the least difficult

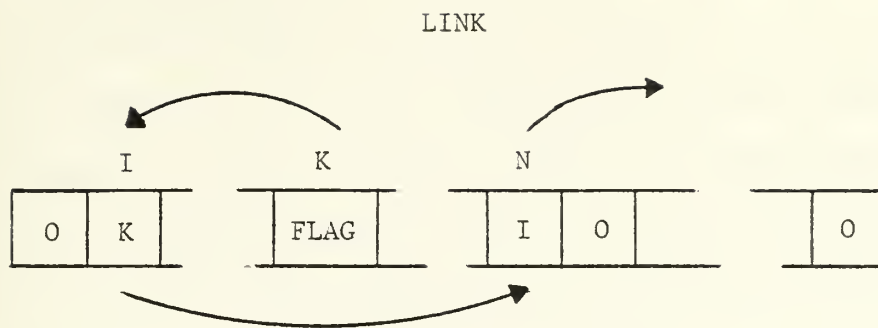


route has been identified. For a network consisting of NNODE nodes, the optimum route will be found after at most NNODE iterations.

Two data lists are required to implement Dijkstra's procedure. One list, described in an article by Pate [Ref. 7], is used to determine whether a node has a permanent or a temporary label. It also links the nodes in reverse order of their occurrence in the routes which have been identified. In the route selection model, this list is called LINK. This list must be zeroed out at the start of each selection process. The second list, called TEMP, contains the node numbers for each node that has been assigned a temporary label. The nodes contained in TEMP are stored in order of increasing difficulty value. An example of how these two lists are structured is provided in Figure 11. Two variables are also required in this procedure. NPERM contains the node number of the last node to receive a permanent label, and DIFF contains the cumulative tactical difficulty for that node.

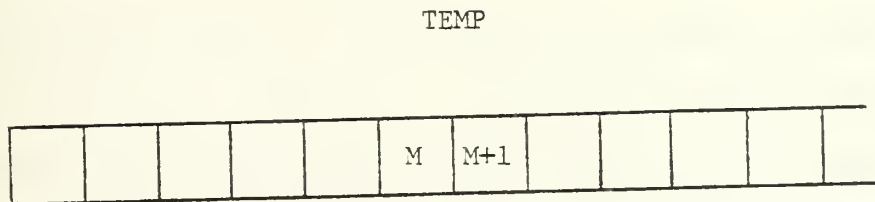
The algorithm begins by identifying the node which represents the maneuver element's current position. This node number is assigned to NPERM, and DIFF is set equal to zero. This node is also identified as the start point of all routes by setting LINK(NPERM) equal to a negative flag value. The list of neighboring nodes of the start point are then obtained from the subroutine NABOR. For each node, the cumulative tactical difficulty of the route leading to that node is computed. This route necessarily includes NPERM as





$\text{NODE}(J)$
 $J = 1, \dots, \text{NNODE}$

{	$= 0$ IF NOT LABELED > 0 IF TEMPORARY LABEL < 0 IF PERMANENT LABEL
---	--



$\text{TD}(\text{TEMP}(M)) < \text{TD}(\text{TEMP}(M+1))$
 $M = 1, \dots$

Figure 11. Shortest-Route Algorithm



the immediate predecessor node to the node being considered. Since this is the first iteration of the algorithm, this cumulative value is just the difficulty for the individual route segment. This difficulty value is calculated using the time and exposure values that have been previously computed and stored. The following equations are used for each neighbor node, $NBR(I)$ for $I = 1, \dots, NNBR$:

$$TD(NBR(I)) = TIME(NPERM(I)) (1 + E(NBR(I))) + DIFF$$

$$LINK(NBR(I)) = NPERM$$

where $TD(NBR(I))$ = temporary cumulative tactical difficulty for the route which terminates at node $NBR(I)$,

$TIME(NPERM(I))$ = travel time from node $NPERM$ to its I th neighbor, and

$E(NBR(I))$ = exposure value assigned to the node $NBR(I)$.

The neighboring node is then stored in $TEMP$ in order of increasing tactical difficulty. Notice that the list $LINK$ has also been updated to reflect that $NPERM$ is the predecessor node in the temporary route to each of the neighboring points.

The next iteration of the algorithm begins by removing the first entry from the list of temporary labels. This node is given a permanent label signifying that the least difficult route from the origin to this node has been identified. This must be true because any other route to this node must include one of the other nodes in $TEMP$. Since the cumulative tactical difficulty for each of these nodes is greater than that of the first node, and since the difficulty values are non-negative, any other route must necessarily

produce a larger cumulative value. This node is permanently labeled by setting NPERM equal to TEMP(1), DIFF equal to TD(TEMP(1)), and LINK(NPERM) equal to -LINK(NPERM). The node's location in the LINK list has been flagged with a negative value to indicate that no other route to that node can reduce the tactical difficulty value at this node.

As before, the neighbors of NPERM are identified and the temporary difficulty values are computed and stored in TEMP. However, two situations can occur which alter this procedure. First, if one of the neighboring nodes is already permanently labeled, there is no need to consider it again because no better route to that node can exist. Therefore, if LINK(NBR(I)) contains a negative value, the computations for that node are skipped. The second situation occurs when the neighboring node is already contained in TEMP. This is the case if LINK(NBR(I)) contains a positive value. In this situation, two competing routes have been identified, and the best one must be selected. The tactical difficulty for the new route, with NPERM as a predecessor node, is computed and compared with the value of the previously identified route. This route has LINK(NBR(I)) as a predecessor. If the new difficulty value is greater than the stored value, the new route and value are disregarded. However, if it is less than the stored value, the new value replaces the old value, and LINK(NBR(I)) is set equal to NPERM to reflect the new route to the node.

This procedure is repeated until the terminal node is given a permanent label. When this occurs, the least difficult route from the initial point to the objective has been identified. The sequence of nodes which comprise this route is obtained by recording the predecessor information stored in LINK, starting at the terminal node and working backwards to the initial node. The sequence is then placed in correct order and the coordinates of these nodes are recalled. The optimum route is now available to the simulation.

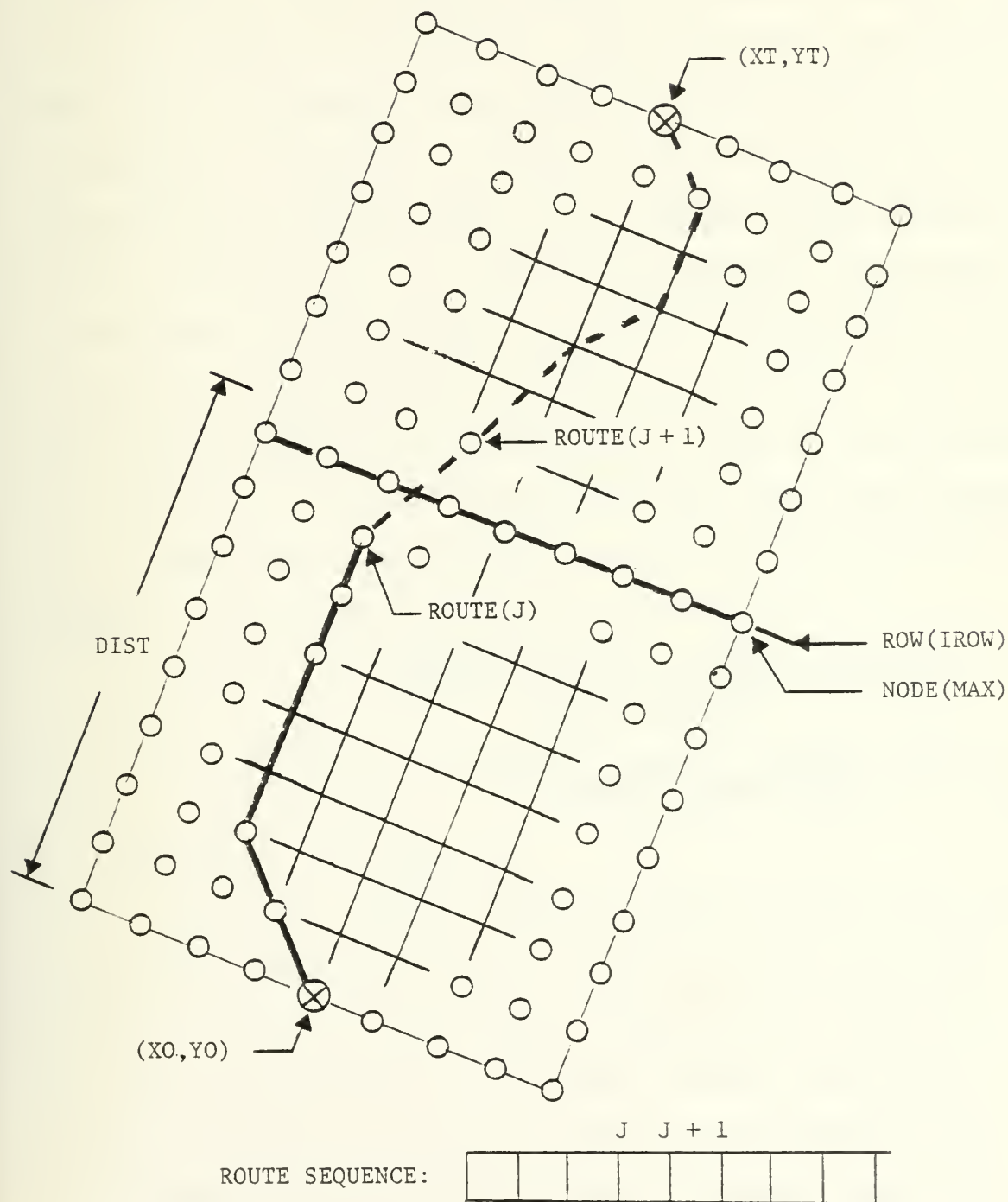
It should be noted that if the selected route is to be re-evaluated after the maneuver element has traveled a specified distance, it is not necessary to provide the entire route description to the combat simulation. Only that portion of the route within a predetermined distance, DIST, of the start point needs to be returned by the route selection model. This concept, and the procedure for identifying the appropriate route section, are shown in Figure 12. However, if the simulation uses an event step procedure, it may be more appropriate to schedule route re-evaluation events at specified time intervals rather than distance intervals. Of course, in either procedure, criteria which reflect the tactical situation may cause a route to be re-evaluated before the time or distance criteria are met.

5. Route Re-evaluation

Whenever it is necessary to re-evaluate a route, a sequence of steps occurs that is similar to that which has

$$IROW = \text{INTEGER} \left[\frac{\text{DIST}}{\text{RFIX}} \right] + 1$$

$$\text{MAX} = IROW \times \text{NCOL}$$



$$\text{ROUTE}(J) < \text{MAX} < \text{ROUTE}(J+1)$$

Figure 12. Model Output



just been described. However, there are some important differences. Rather than establishing a new route selection array based on the maneuver element's current position, the original array is maintained. This eliminates the requirement for re-computing array parameters, node numbers, and coordinates. However, the element's current position will typically not coincide with any of the previously established nodes. Therefore, it is necessary to identify the node which is nearest to this position. This is accomplished by transforming the coordinates of the current position, (X,Y), to the corresponding array coordinates. These I-J coordinates are rounded to the nearest integer values and then converted to a single node number which is stored in the variable NSTART. The following equations are used:

$$I = \text{Round} \left[1 + \frac{(XT-XO) (X-XO) + (YT-YO) (Y-YO)}{(NROW-1) RFIX^2} \right]$$

$$J = \text{Round} \left[JHALF + \frac{(YT-YO) (X-XO) - (XT-XO) (Y-YO)}{(NROW-1) RFIX \ CFIX} \right]$$

$$NSTART = (I-1) NCOL + J$$

The node labeled NSTART is used to define the route selection area to be used for the re-evaluation process. Figure 13 shows how this area is identified. Notice that the area includes the row of nodes "behind" the current position. This is to allow a route to be selected which can attempt to circumvent an enemy influence. Although only one row of nodes is provided for this purpose, the concept can be extended to include any fixed number of additional rows. Also notice that the initial set of potential route

$$\text{MIN} = (I - 2) + 1$$

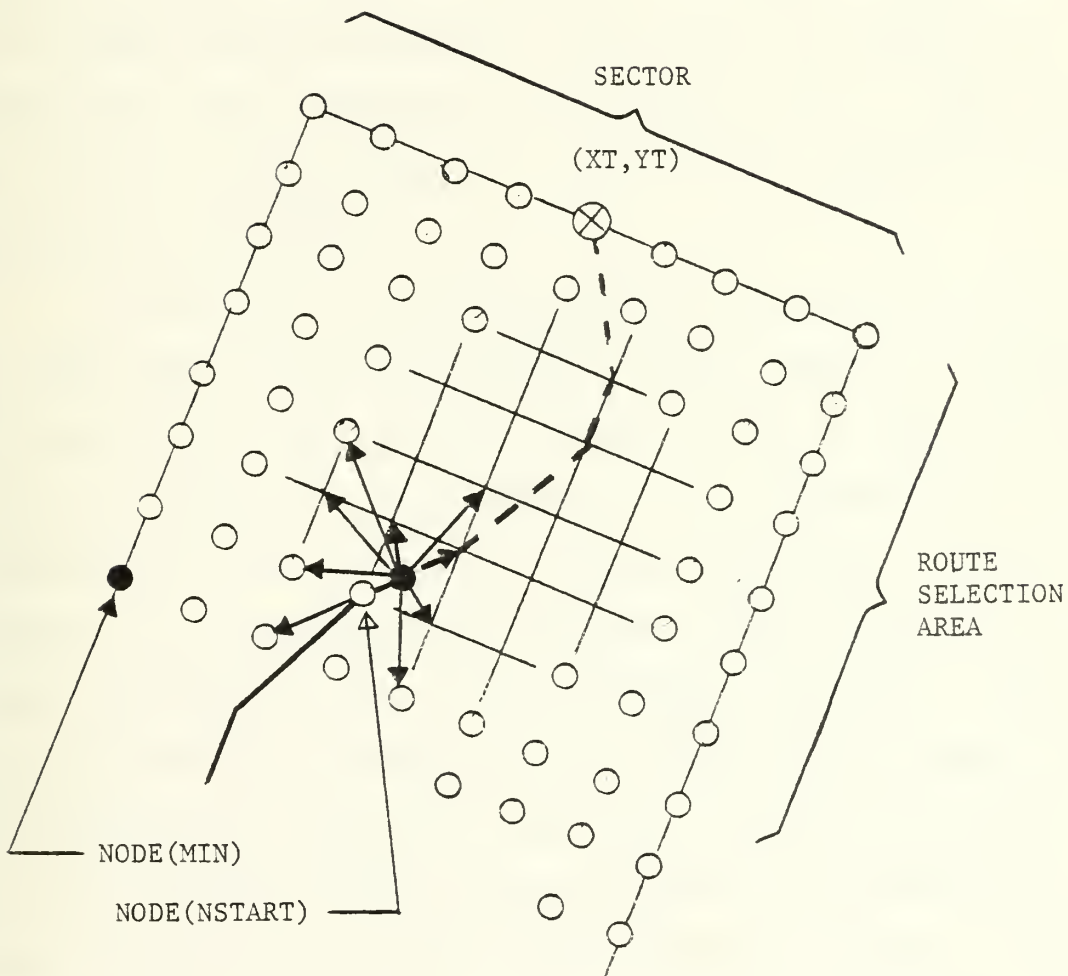


Figure 13. Route Re-evaluation



segments radiates from the element's current position and not from the node labeled NSTART. To account for the difference in locations, the coordinates assigned to this node are replaced by the coordinates (X,Y). In addition, the set of travel times from NSTART to its neighbor nodes is replaced by the travel times from the actual starting location to the same set of neighbors. Thus, the node NSTART has essentially been transposed to the point (X,Y). This allows the original node number to be used in all subsequent route evaluation procedures.

Within the new route selection area, which consists of those nodes numbered from MIN through NNODE, only those influence factors which have changed since the last route was selected need to be considered. If a factor which influences route selection has remained unchanged, its contribution to the exposure weights and travel times is already reflected in the values previously computed. Those factors which have changed can be identified by comparing the information contained in the lists STATUS and CHANGE. The list CHANGE contains those temporary influences which were included in the last route evaluation process. STATUS contains those influences which represent the maneuver element's current impression of the tactical situation. In Figure 14, the four general types of changes that can occur are presented. The procedure used to update CHANGE to reflect these events is also shown in the figure. The flag values which are used in this procedure are initially set

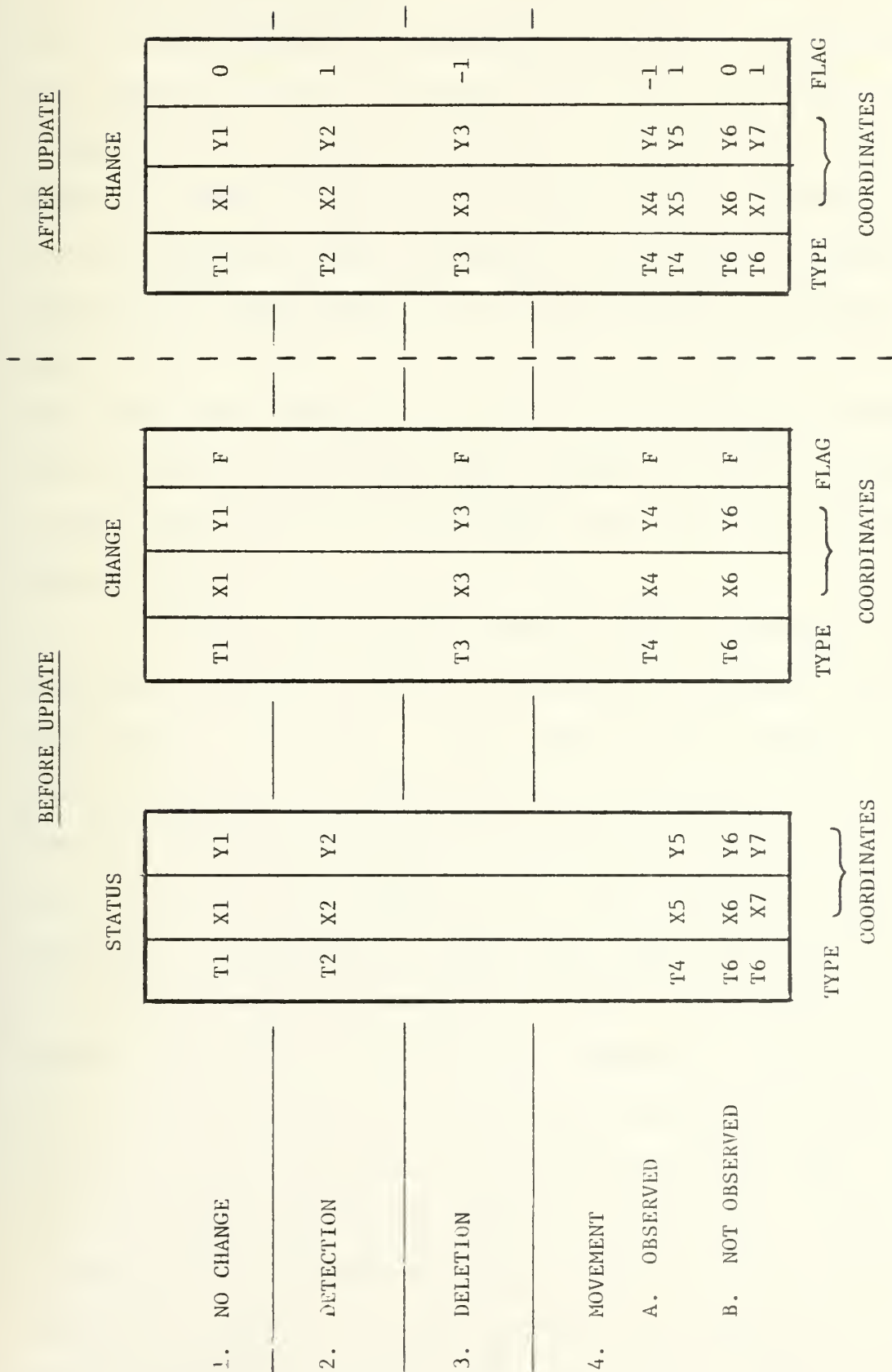


Figure 14. Update of CHANGE



to a predetermined value. For each item in STATUS, the CHANGE list is then scanned to determine if the type and location of that item are already listed. If so, the flag value is set to zero. If it is not contained in the list, the item is added and given a flag value of positive one. After every item in STATUS has been evaluated, any entries in CHANGE which still retain the initial flag value are assigned a flag of negative one. At this point, the CHANGE list has been completely updated. The flag values indicate whether the influence factors have been deleted, remained constant, or been added (by the values minus one, zero, positive one respectively).

The existing exposure factors for the nodes in the route selection area can now be altered to reflect the current tactical situation. Since these values already reflect the influence of the fixed elements contained in the SUSP list and also those temporary elements which possess a flag value of zero in the CHANGE list, these items need not be considered. Therefore, at each node, the subroutine EXPOSE is called to compute an exposure weight for each element in CHANGE which has a non-zero flag. However, in the route re-evaluation process, the individual weights which are returned by this subroutine are multiplied by the flag value assigned to the respective sources of influence. These weights are then summed and added to the existing value assigned to the node. By this process, the appropriate exposure weights are either subtracted from or added to the

existing value, depending on whether the source of influence has been deleted from or added to the maneuver element's representation of the tactical situation. This process is repeated for those nodes which comprise the current route selection array. Of course, those nodes which have been previously flagged to indicate that they are within assault range of the objective need not be considered in this updating procedure.

The previously described route selection process can now be employed to select a new route from the NSTART node to the objective. This new route, or a portion of it, is then provided to the combat simulation. This route re-evaluation process is successively applied, as the need arises, until the maneuver element is sufficiently close to the objective.

C. IMPLEMENTATION OF THE MODEL

In the previous section, the structural and computational aspects of route selection model were presented. These aspects were concentrated within a limited scope established by the objectives of the thesis. There are, however, two general subject areas that deserve additional attention. Although not critical to the functions within the model itself, these subjects will become important to the capabilities of the model when it is integrated into a combat simulation.

The first area concerns the information flow from the simulation to the model. The three main categories of



information that the model requires are line of sight, travel time, and tactical intelligence. The line of sight function poses no real problem because this information is required only in the programmer's EXPOSE subroutine. The subroutine format can be easily adapted to match the specific format of the function. Although the model requires that the computation of travel time be structured in a rather specific manner, this is necessary if dynamic route selection is to be included in the simulation. However, in the area of tactical intelligence information, a great deal of flexibility is allowed by the model. The types of tactical situations that can be represented in the route selection model are determined by the capability of the simulation to monitor the details of the situation and to record the appropriate data in the array called STATUS. The information that is required for the four types of events described in Figure 14 deserves additional discussion.

The detection and no-change events require only that the simulation have the capability to report the location and type of the detected influence factors. However, an element can be deleted from STATUS in two ways. If the element has been destroyed, it should be deleted from the list immediately. If detection has just been lost, the element may also be removed from the list. However, in this case, it seems realistic to assume that, although the detection has been lost, the element still remains in the area of the previous location. To reflect the maneuver element's impression of the tactical situation, this influence factor should



be retained in STATUS for a specified time period. A similar situation exists when an enemy weapon system changes locations. If the maneuver element observes the enemy movement, the new location of the enemy should replace the previously recorded location in the list. However, if the movement is not observed, this situation should be treated as a loss of detection at the previously recorded location and as a detection at the new location. Thus, two enemy influences should be represented in this case.

It should also be noted that the STATUS list is designed to reflect all temporary influences that affect the route selection process. This includes not only enemy weapon systems but also minefields, obstacles, and natural barriers. The term "temporary" is used here to include any such items which are not initially known to the maneuver element and may be encountered during movement. Again, the specific types of influences and the method of representation depend on the structure of the simulation. The calculations which are required to quantify these types of influences seem most appropriately handled within the EXPOSE subroutine. These calculations might involve not only exposure weights but also travel times. For example, a detected minefield will affect both travel time and exposure for those nodes within the minefield. If soil types are represented, a detected swamp, for example, might only affect travel time. The subjective decision as to how these influences affect either exposure or time is left to the programmer. However, for



those influences of fixed size and location, it would be efficient to identify all nodes affected by the influence and to compute the changes for this set of nodes rather than to evaluate each node in the entire array to determine if it is affected by this influence. This can be accomplished when the first node in the route selection array is evaluated. The flag assigned to the influence factor in the CHANGE list can then be set to zero to indicate that the affects of this particular factor have already been computed.

Regardless of the types of tactical influences which can be represented in the STATUS list, the procedure used in the route selection model to update the CHANGE list will correctly identify any change. Of course, the EXPOSE subroutine must then be capable of processing these changes into appropriate time or exposure values.

The representation of the tactical situation should also be considered in determining the criteria which specify when a route needs to be re-evaluated. It seems reasonable to assume that a maneuver element might consider a new route whenever it is fired upon. It would also seem reasonable to re-evaluate a movement route after the number of changes that occur in the STATUS list reaches a specified level. Together with the fixed travel time or distance increment, these criteria need to be defined within the simulation. However, there should also be a criterion for by-passing these re-evaluations whenever the maneuver element gets reasonably close to the objective.



The second major area of application involves the identification of the route selection area. At each iteration of the route selection process, a route is generated which terminates at the point which has been identified as the objective. Thus far, the concept of an objective has been used in the tactical sense of the word. This is defined to mean a piece of terrain which a tactical element seeks to seize and physically occupy. However, it may be convenient to allow an alternate form of termination point for the route. Instead of an objective, a control point might be specified. This control point may represent a movement restriction, a turning point within the sector of advance, or an intermediate piece of terrain that the element is required to pass through. The use of control points can serve to reduce the computer storage required by the route selection array. For example, if the sector is relatively large, there will be a large amount of data that must be stored and manipulated. This storage requirement can be reduced by identifying a series of intermediate control points within the sector. The model will then successively treat each control point as a termination point for a smaller route selection array. This type of procedure will produce a total route that has been identified by a suboptimization process.

If a control point is used, it must be dealt with differently than if it were a tactical objective. In general, the concept of an assault range is not applicable to control points. Therefore, the route leading to this point should be

optimized in terms of both time and exposure throughout the entire route selection area. Also, since control points might be used to approximate sections of an optimum route which ordinarily would not pass through these exact points, the array for a succeeding route section should be generated before the tactical element actually reaches the terminal point in the current array. This allows a smoother transition into the succeeding route selection area, and does not necessarily force the route to pass over the precise control point.

An alternate concept can be used to describe the terminal point for any particular array. In addition to specifying the coordinates of the terminal point, the programmer could also specify a width for a terminal area. Rather than requiring that the route end on one specific point, this would allow the route to lead to any node in the last row of the array which is within the specified half-width of the terminal point. This would provide an additional smoothing technique for transitions between route arrays. In addition, since a tactical objective or a route restriction is typically an area rather than a point, this concept could be used to represent these areas.

Certainly, other subjects of concern exist in the area of implementation of the model. However, the two concepts that have been discussed in this section seem to exert a major influence over the potential capabilities of the route selection model.

V. DOCUMENTATION OF TEST RESULTS

The dynamic route selection model that was developed in Chapter IV has been programmed, and the organizational and computational aspects of the model have been verified. A source listing of the FORTRAN program is contained in Appendix C. In addition, an alphabetical list of the major variables and their definitions is presented in Appendix B. In this chapter, the test situations that were used to exercise the model are documented. The routes which resulted from these tests are also presented and discussed.

In order to realistically exercise the route selection model, components of the STAR combat simulation were used to provide input data for the model. However, in order to control the test situations, the stochastic portions of the simulation were not utilized. Only the terrain, movement, and line of sight models, with their supporting subroutines, were used. The remainder of the input data were provided as fixed input through the use of a main program which served to represent the typical information flow that would exist between the model and the complete simulation. In addition, for both the movement model and the line of sight model, the existing calling sequences were simplified for the sake of clarity. The movement model was substantially modified in order to provide information in the format required by the route selection model. Originally, this movement subroutine



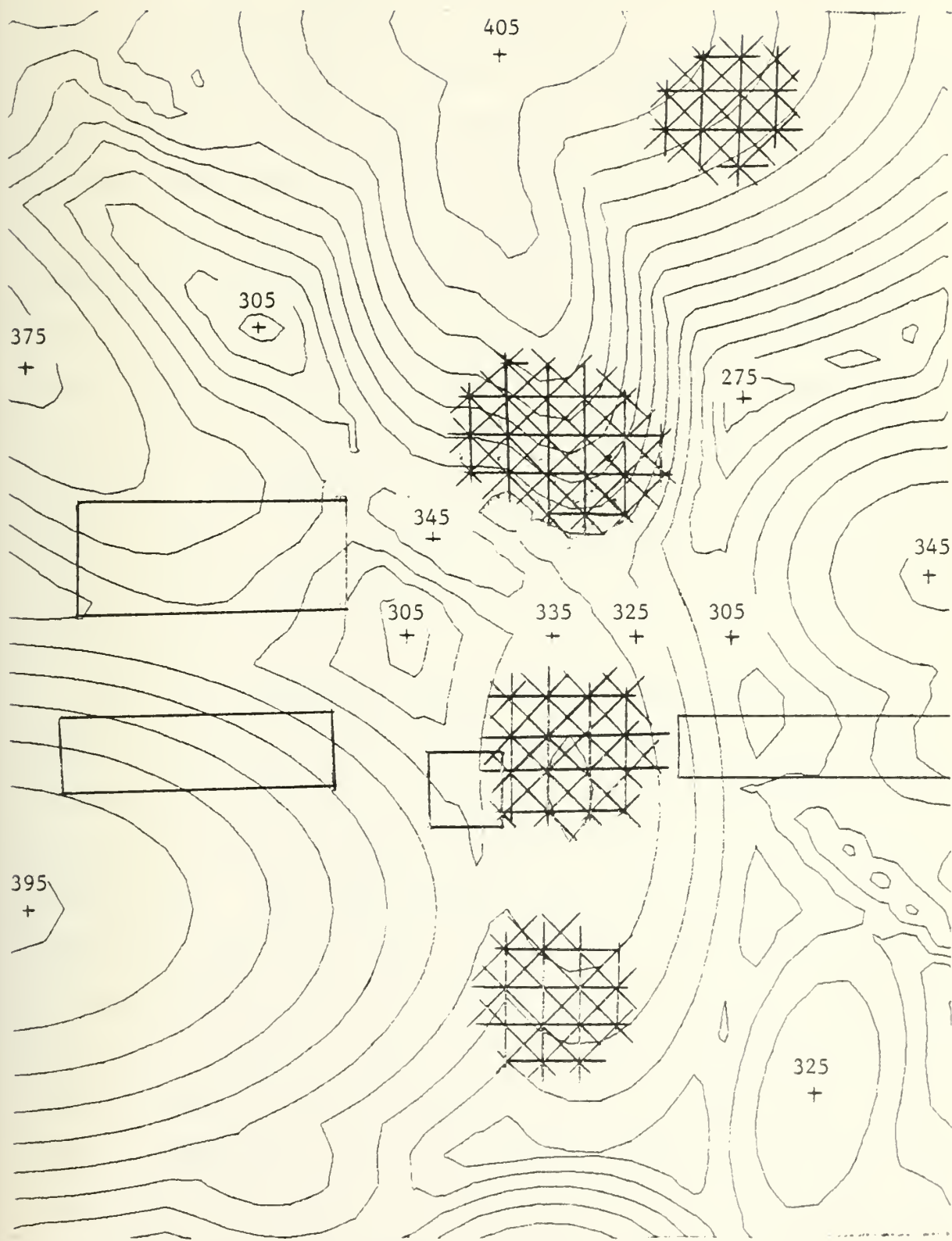
was designed to accept as input the current position of the maneuver element and a time increment. The model would then compute the position on a specified, fixed route where the element would be located at the end of the time increment. This procedure was altered to allow the movement model to accept the coordinates of two points, which define a route segment, and to compute the travel time between these points. The section of the movement subroutine dealing with minefields and their impact on travel time was also removed from the model for these tests. Finally, the movement model was renamed TIME, again, merely for clarity.

For the tests that were conducted, the following input parameters were specified. Desired row and column spacings for the route selection array were set at 100 meters. The width of the movement sector was fixed at 1000 meters, and an assault range of 600 meters was used. The number of allowable neighboring points was set at nine, and the pattern of routes that was shown in Figure 10 was used. Three types of enemy influences were portrayed: suspected enemy positions were assigned a relative weight factor of three units and a range of influence of 3000 meters, enemy tanks were assigned a weight of six and an effective range of 2500 meters, and enemy anti-tank systems were assigned weights of four units for an Anti-Tank-Guided-Missile system and two units for a smaller caliber weapon with respective ranges of 3000 and 1500 meters.



In the subroutine EXPOSE, exposure weights were computed as a linearly decreasing function of range. In addition, line of sight computations were used to contribute an additional weight of one unit if intervisibility existed. Since the line of sight model that was used computed the fraction of the moving vehicle that is visible to the enemy element, a value of ten percent was selected to determine if intervisibility existed. If less than ten percent of the vehicle was visible, no line of sight weight was applied. The full weight of one unit was added if more than ten percent of the vehicle was visible to the enemy element.

From the terrain represented in the STAR model, an area was selected which provided a representative sample of the various factors portrayed in the simulation. The area of terrain that was chosen is shown in Figure 15. In the figure, the rectangular shapes represent areas of highly reduced mobility. The cross-hatched areas represent forests. It should be noted that the forest areas do not hinder the mobility which is determined by the subroutine TIME. These areas do, however, significantly affect intervisibility. In addition, the slope of the terrain at any point is computed in TIME prior to the calculation of travel time. However, the slopes are first classified into one of three categories: down slope, level, or up slope. Therefore, only significant changes in slope actually influence the computation of travel time. However, considering the relatively gentle terrain depicted in Figure 15 and the mobility



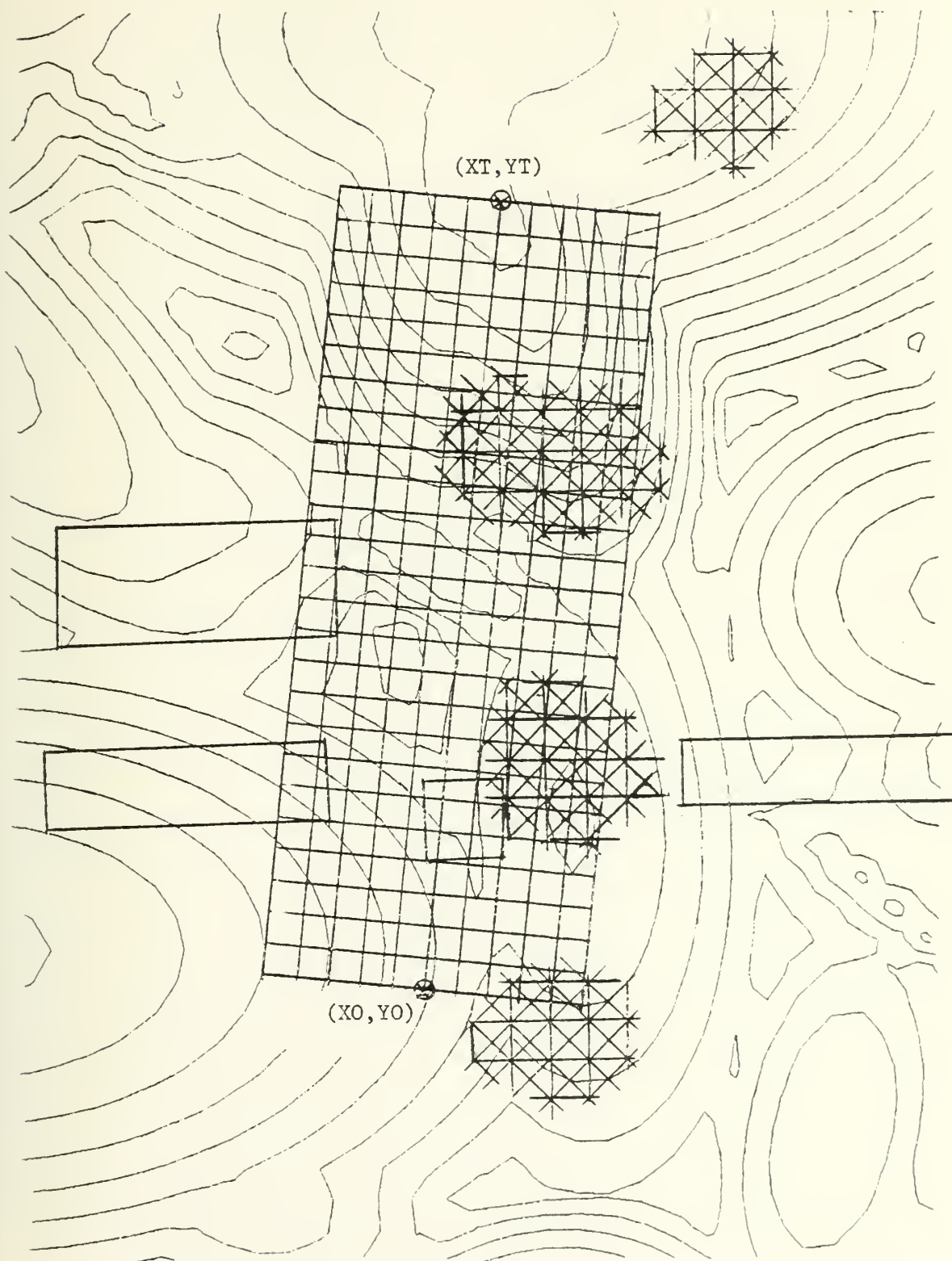
SCALE 0 1000 METERS

Figure 15. Sample Terrain

of a typical mechanized vehicle, this form of slope representation is not totally unreasonable.

Within this area of terrain, an initial point (XO,YO) and and objective (XT,YT) were specified. The route selection model then generated the array of grid points shown in Figure 16. It should be noted that the position of the array was not selected to represent a realistic tactical situation, but rather to include terrain which could be used to exercise the various aspects of the model. Based on the input parameters and on the distance between the initial point and the objective, the depicted array consists of 26 rows, 11 columns, and 286 nodes. As a result of the rounding process, the actual row spacing is 100.5 meters, and the actual column spacing is 100.0 meters.

The route selection model was then used to determine a route of minimum travel time to the objective. This route is shown in Figure 17. Due to the simple classification of slope, the sample terrain area was treated essentially as a level plane. Thus, the optimum route that was selected is basically a straight line. It has been deflected slightly to the left to avoid the area of reduced mobility. For this particular terrain, except for the areas of reduced mobility, travel times for route segments of equal length were exactly equal. Therefore, there are actually several optimum routes available. The route could have returned to the center line of the array at any grid point beyond the one area of difficulty. Any such route would have resulted in a total travel time equal to that of the route which was actually selected.



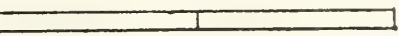
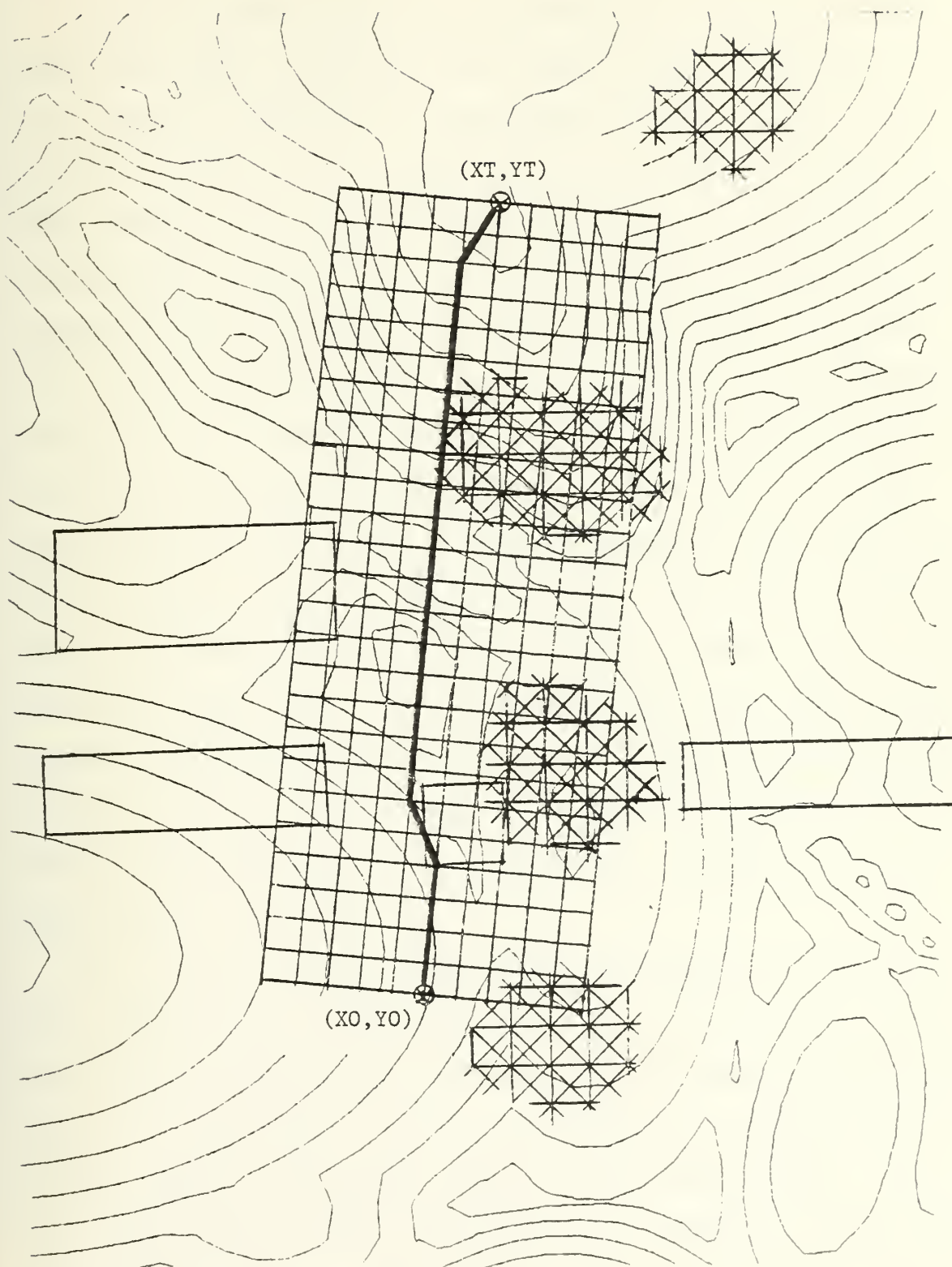
SCALE  METERS
0 1000

Figure 16. Example Array





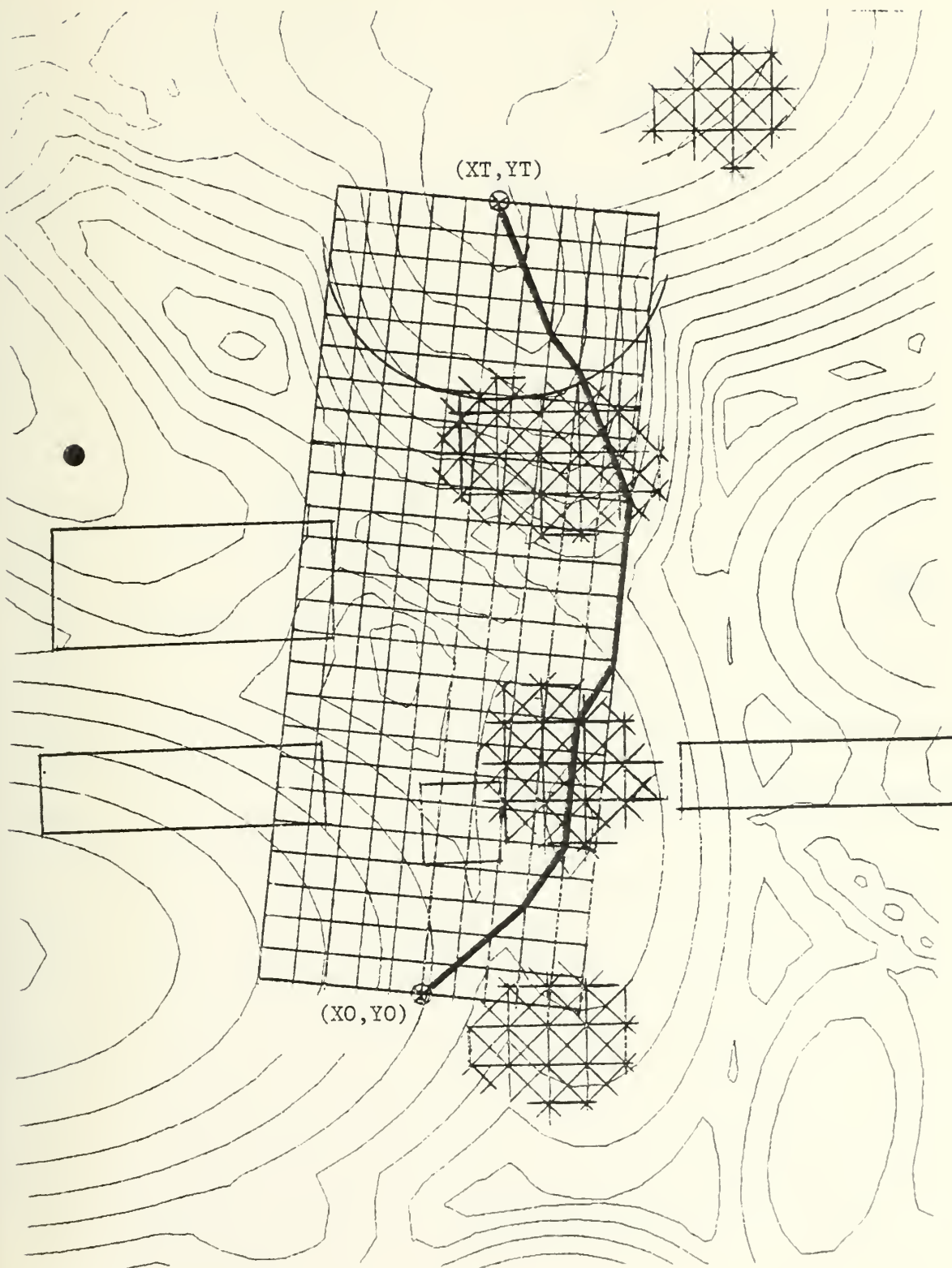
SCALE 0 1000 METERS

Figure 17. Least-time Route



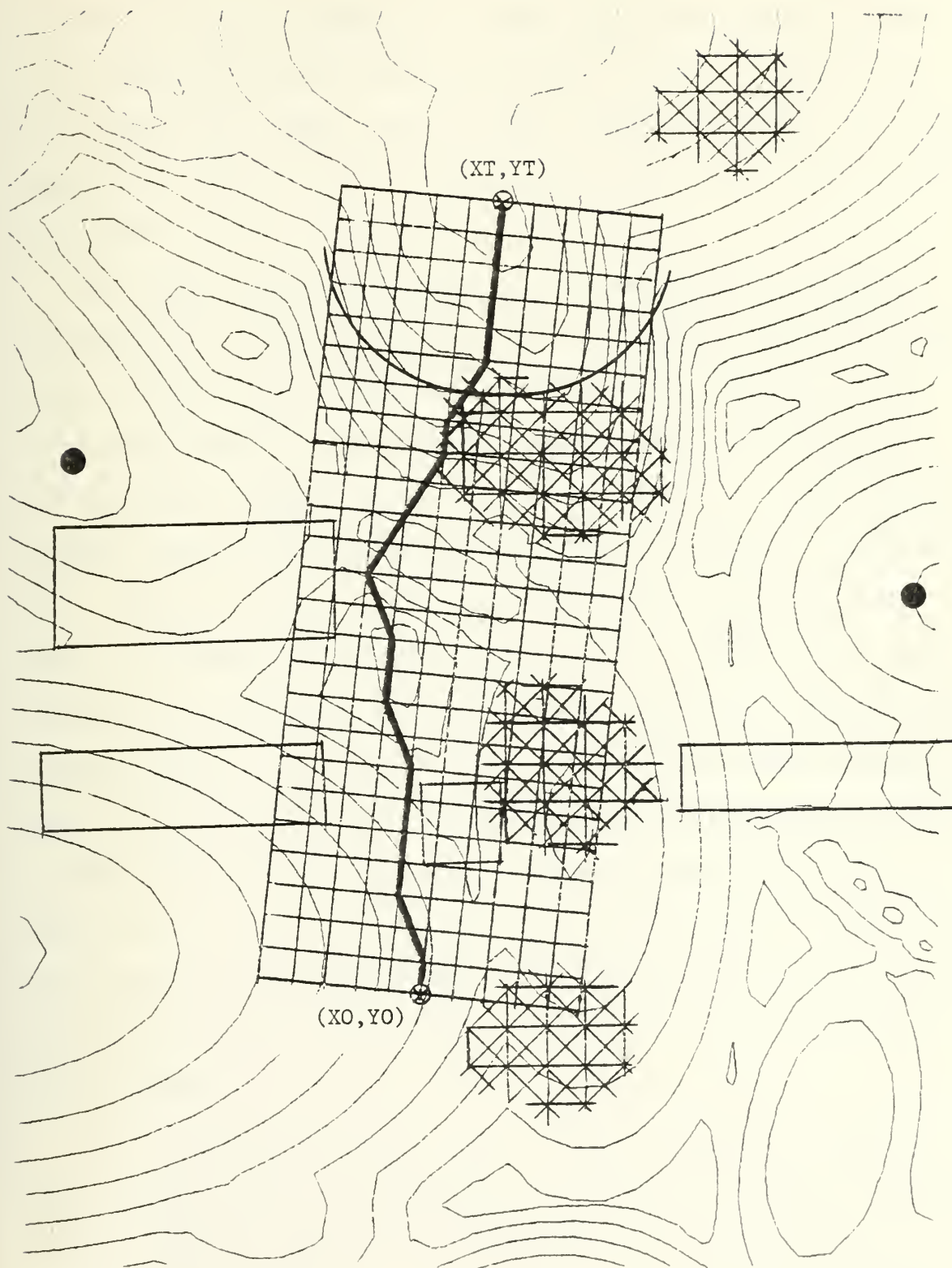
Initial routes were also generated in situations where suspected enemy influences were represented. In Figure 18, one suspected location was positioned outside of the left boundary of the sector. Because of the range of influence (3000 meters), the entire route selection area was under the influence of this position. The decreasing exposure function forced the route to the right side of the sector. In this area, line of sight became the critical factor in the actual location of the optimum route. As can be seen in the figure, the route that was selected passed through the two forest areas where line of sight from the suspected enemy position did not exist. Between these two areas, the route followed the low ground along the right boundary of the sector. Had the sector been wider, the route probably would have passed through the small valley just outside of the current sector. Beyond the last forest area, the route came within assault range of the objective, and travel time dictated the subsequent section of route.

In a second initial situation, two suspected positions were represented as shown in Figure 19. In this case, the two exposure functions forced the route to pass through the center of the sector. Again, intervisibility played an important part in the positioning of the optimum route. The fact that the two enemy positions were located on relatively flat hill tops, restricted line of sight to that portion of the route located in the left center of the sector where a depression exists. Beyond this low area, the route passed



SCALE 0 1000 METERS

Figure 18. Initial Route (A)



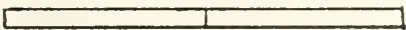
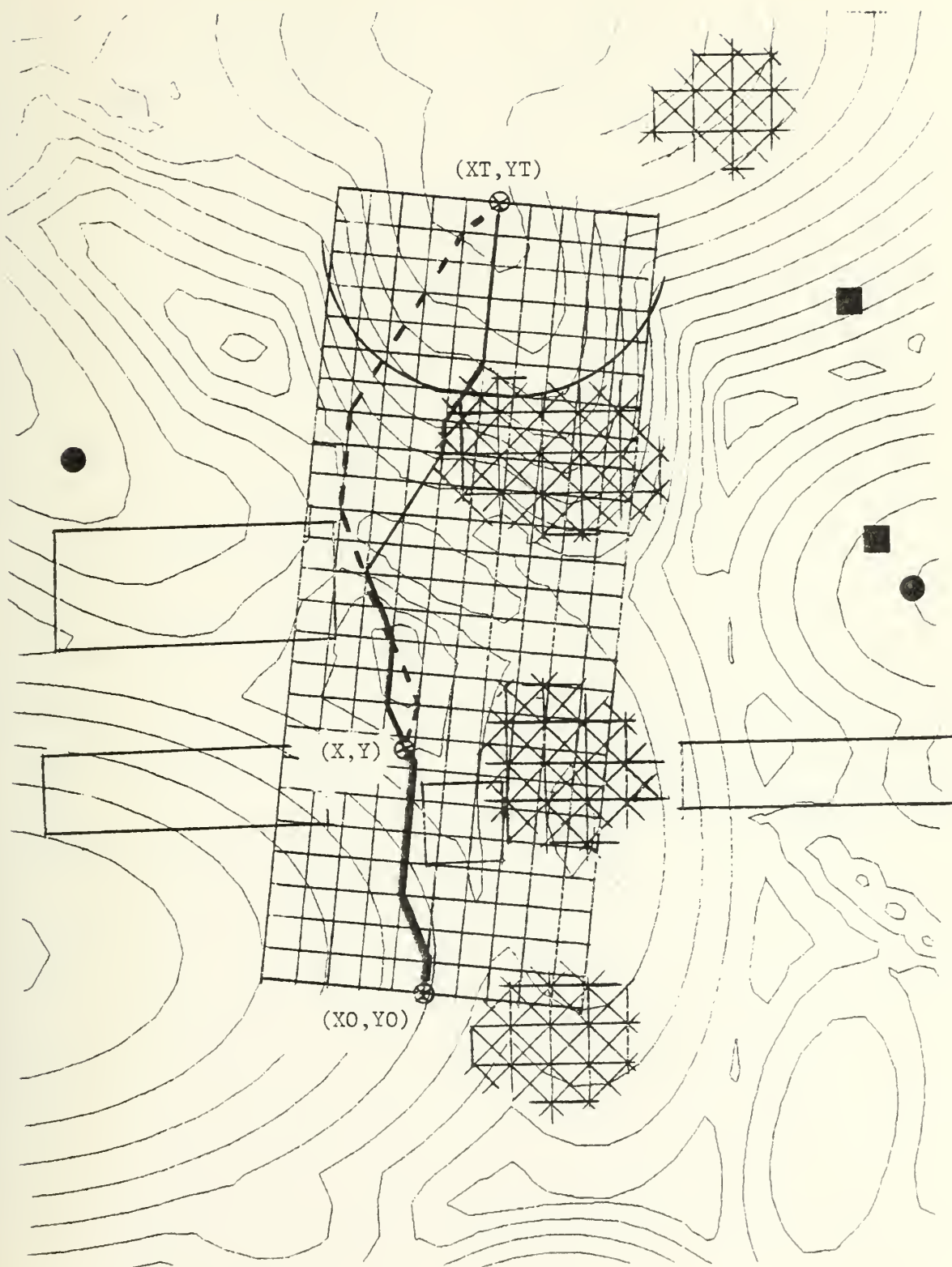
SCALE  METERS
0 1000

Figure 19. Initial Route (B)

to the left of the small hill mass to avoid line of sight from the position to the right of the sector, and through the edge of the tree line to avoid intervisibility with either position. Again, travel time dictated the final section of the route.

The above tactical situation was expanded to include a re-evaluation of the initial route. The new tactical situation and the resulting route are shown in Figure 20. This figure represents a maneuver element currently located at the position labeled (X,Y). At this point, two enemy weapon systems were detected beyond the right boundary of the sector. Based on this situation, a new route was generated from the element's current position to the objective. The new exposure and intervisibility considerations altered the route slightly in the area near (X,Y). The larger weights from the detected elements dominated the suspected position on the left side of the sector. This forced the route toward the left side of the route selection array. There, the hill mass and the forest served to reduce the total relative exposure by eliminating line of sight with the new sources of influence.

The tactical situation was further expanded by conducting an additional re-evaluation of the most recently computed route. The new tactical situation and the resulting route are shown in Figure 21. This figure represents a maneuver element that has advanced along the previous route to a new position labeled (X,Y). In addition, a new source of



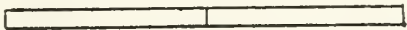
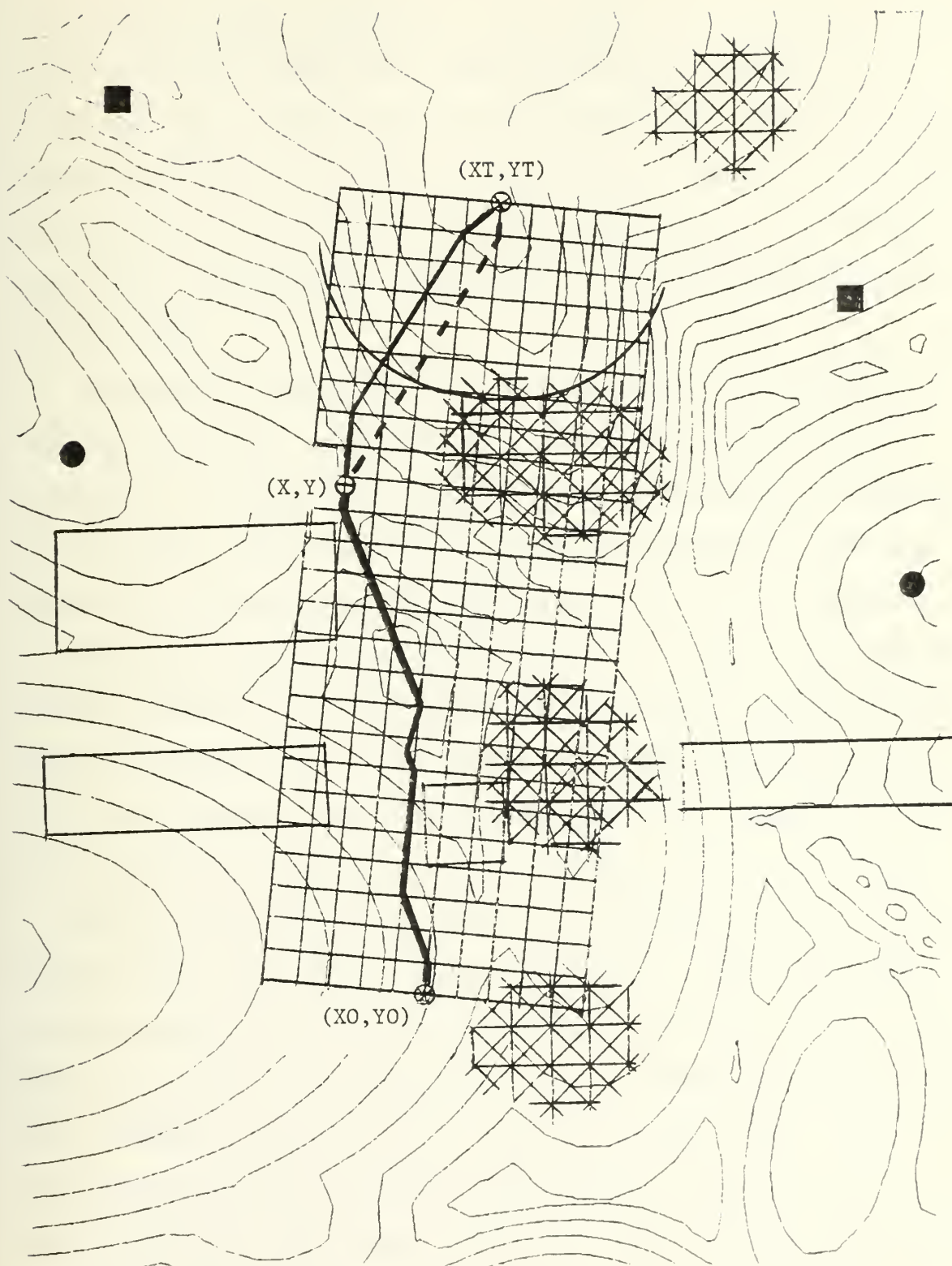
SCALE  METERS
0 1000

Figure 20. First Route Re-evaluation





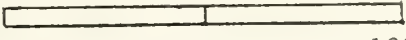
SCALE  METERS
0 1000

Figure 21. Second Route Re-evaluation



influence has been detected in the upper left corner of the figure, and the previously detected weapon system at the right center of the figure has been deleted. The resulting route was shifted slightly to the right. However, the close proximity of the route to the objective resulted in travel time being the dominant influence in the final positioning of the route.

Although no additional major situations were evaluated, these limited test runs have provided the data that was necessary to verify the structure and computational aspects of the model. They have also served to provide an initial indication of the computer execution time that is required to implement dynamic route selection in a combat simulation. For the specific size and shape of the route selection grid that was used in the trial runs, the total computer execution time ranged from 35 to 40 seconds. This time includes all computations required in one initial call to the model.

By far the largest portion of the computer time was devoted to the calculation of travel times. Again considering the specific grid that was used, an initial call to the model required 2275 subsequent calls to the subroutine TIME and consumed a total of 31.6 seconds of CPU time. The computation of exposure weights accounted for most of the variability in the total execution time. The resources required to produce these weights depends not only on the number of sources of influence that are represented in the initial tactical situation, but also on the number of nodes that

are within assault range of the objective. No exposure calculations are required at these nodes. In addition, the time consumed in these calculations by the line of sight subroutine depends on the specific terrain that exists between any two specific points. In the initial tactical situation that was last described, where two suspected locations were represented and assault range was set at 600 meters, a total of 452 calls to the line of sight subroutine were required. This required slightly more than three seconds of computer time. Execution time for the dynamic programming algorithm consistently remained below 1.5 seconds for all initial routes that were generated.

The execution time required to re-evaluate a route is highly dependent on the size of that portion of the grid which comprises the new route selection area. Again, the number of changes in the tactical situation and the length of the assault range also influence the time. For the two re-evaluations which were conducted in the last example, approximately four and three seconds were required respectively. However, these times reflect the fact that no sources of influence were represented which would require the re-calculation of travel times. Had minefields or obstacles been included, the computer time would have increased significantly. In the re-evaluation situations which were described, the only calculations of travel time that were required were those necessary to transform the node nearest to the maneuver element's current position to the coordinates

of that current position. In each situation, this resulted in nine calls to the subroutine TIME. The average time per call in both situations was almost identical to that required for the initial route selection. Based on the row and column spacings within the route selection grid, it appears that approximately 70 complete time calculations per second can be achieved. The execution time required for line of sight computations was again variable. However, an average of roughly 140 calls per second was obtained. This is based on the placement of the sources of influence on dominant terrain features where intervisibility typically existed with most nodes in the route selection grid. Had these sources been positioned in less prominent locations, individual line of sight calculations would have required less time since the calculations are terminated as soon as the first barrier to intervisibility is identified.

In the test situations that have been used to demonstrate the route selection model, the resulting routes appear to be quite reasonable. They can be easily justified based on the terrain and on the specific tactical situations. However, these tests were conducted to verify the internal functioning of the model. No analysis was conducted to verify the appropriateness of the subjective exposure weights and ranges that were utilized. These parameters, along with the tactical difficulty objective function, were specified solely for the purpose of exercising the model.



VI. CONCLUSIONS AND RECOMMENDATIONS

The conclusions that can be drawn from the analysis of the route selection process and from the limited test runs of the route selection model are discussed in this chapter. In addition, recommendations for future study and expansion of the model are presented. The justification for the decision to use a discrete grid concept in the representation of the route selection process has been discussed in Chapter III. However, the results of the initial experiments which were described in the last chapter serve to validate the capability of this approach to be used in the selection of optimum tactical movement routes.

The tests also indicate that the route selection model is relatively expensive to implement in a combat simulation due to the computer time required to support the model. However, since the largest portion of the time is consumed in the computation of travel times, the requirement for computer resources could be reduced by decreasing the number of allowable route segments which emanate from each node. The advantage achieved by this approach would be at the expense of flexibility and smoothness in the resulting routes. Additional effort should be devoted to an analysis of this trade-off.

A limited evaluation of the sensitivity of the route selection model to the assignment of exposure weights for

the various sources of influence was conducted. It appears that the magnitudes of these weights do not significantly affect the location of the final route. However, the relative weights assigned to these different sources of influence do affect the location of the optimum route. Since the weights that were used in the trail runs were selected merely for demonstration purposes, an additional study should be made in order to identify appropriate weight factors and also appropriate ranges of influence.

The sensitivity of the model to the slope of the terrain was not able to be evaluated because of the procedure used to classify terrain in the TIME subroutine. A more detailed representation of the affect that the gradient of the terrain has on the travel times could, in specific situations, significantly alter the location of the optimum route. However, a more sophisticated movement model would undoubtedly require additional computer resources, and it is not clear that the additional detail that would be provided would justify the increased consumption of resources. In addition, since the travel times used in the model are only intended to represent estimated values, a less detailed TIME routine may be more appropriate. This routine could be designed to provide a cruder time estimate, but one that is more representative of the terrain gradient. This would serve to provide a more realistic representation, and, at the same time, could possibly reduce execution time.

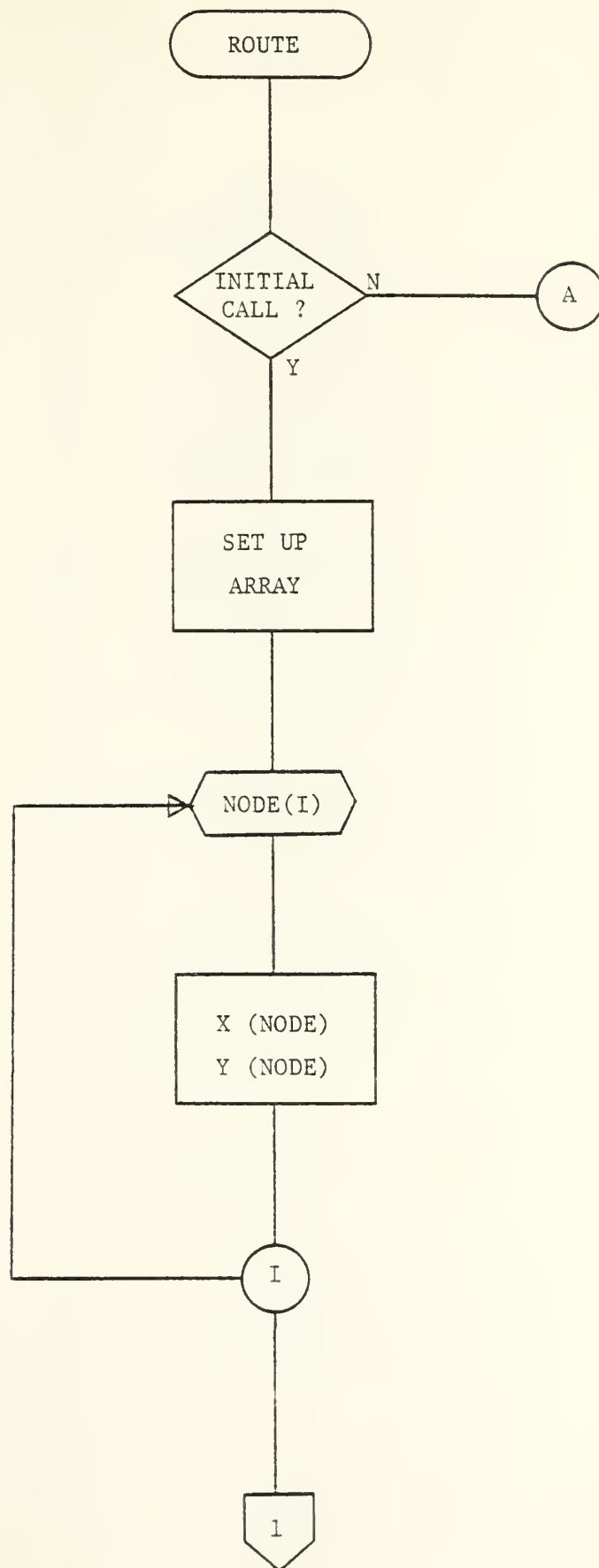
The tactical scenario that has been modeled in this thesis is rather limited. An obvious future expansion of the model would be to modify the structure in order to represent the selection of a route for a tactical unit rather than a single maneuver element. In addition, the model could be expanded to be capable of accepting, and appropriately processing, a sequence of objectives or control points. It also seems logical to include a capability for generating and storing route selection data for a set of maneuver units which are moving through the battlefield simultaneously.

These potential areas for future expansion do not necessarily exhaust the capabilities of the route selection model. They do represent a substantial requirement for analytic and programming effort. However, these concepts seem to encompass the major steps that would be necessary to completely implement the model in a combat simulation.

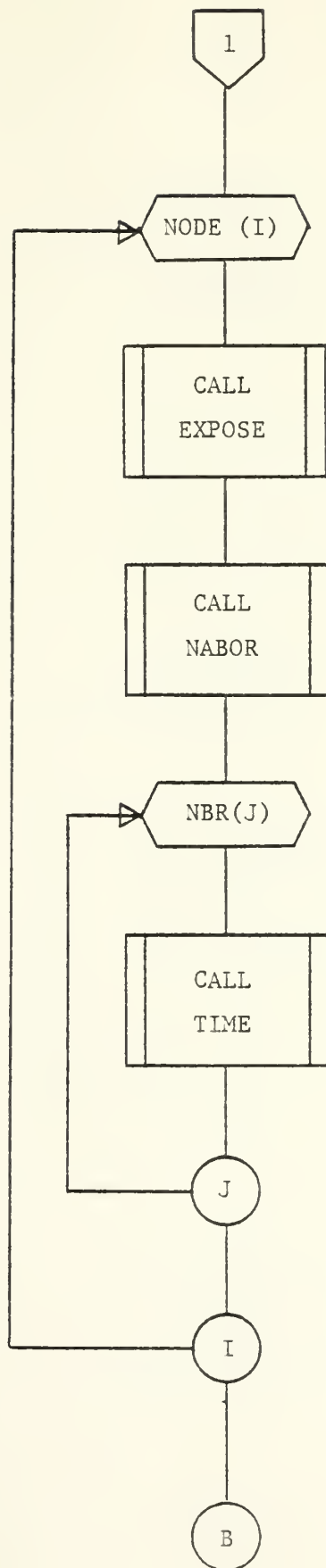


APPENDIX A. FLOW CHART

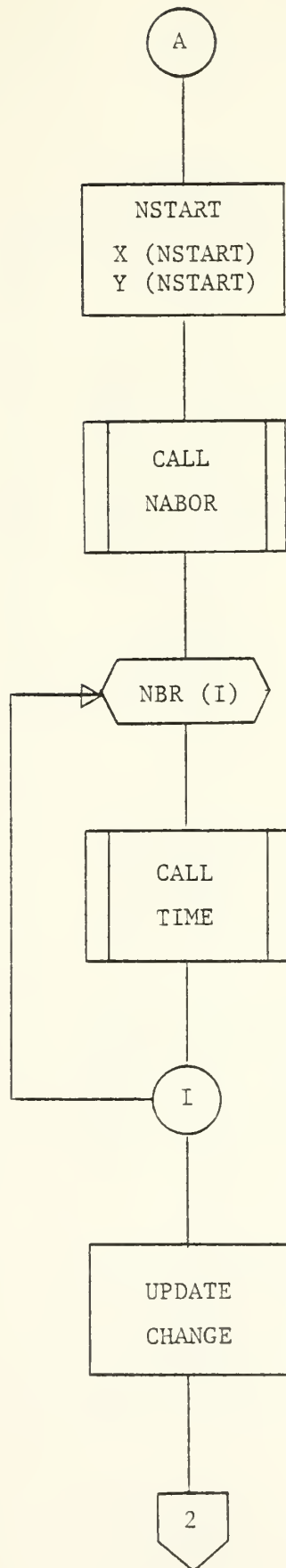
This appendix provides a flow chart of the structure and major processes that are contained in the route selection model. The primary subroutine ROUTE is depicted as well as the subroutine SELECT which performs the optimization process. The NABOR subroutine is not contained in this appendix because of its relative simplicity. A detailed description of this routine is contained in Chapter IV. In addition, the programmer prepared subroutine EXPOSE is omitted. However, one example of this routine is contained in the program listing which is presented in Appendix C. Finally, the supporting subroutines from the parent combat simulation are referenced only by the appropriate calling statements. The particular requirements for special structure and format of these routines are addressed in Chapter IV.

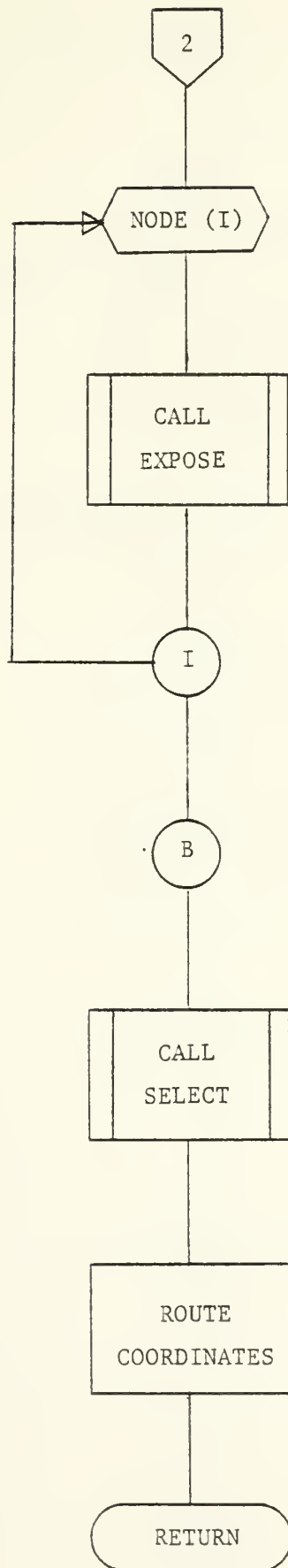




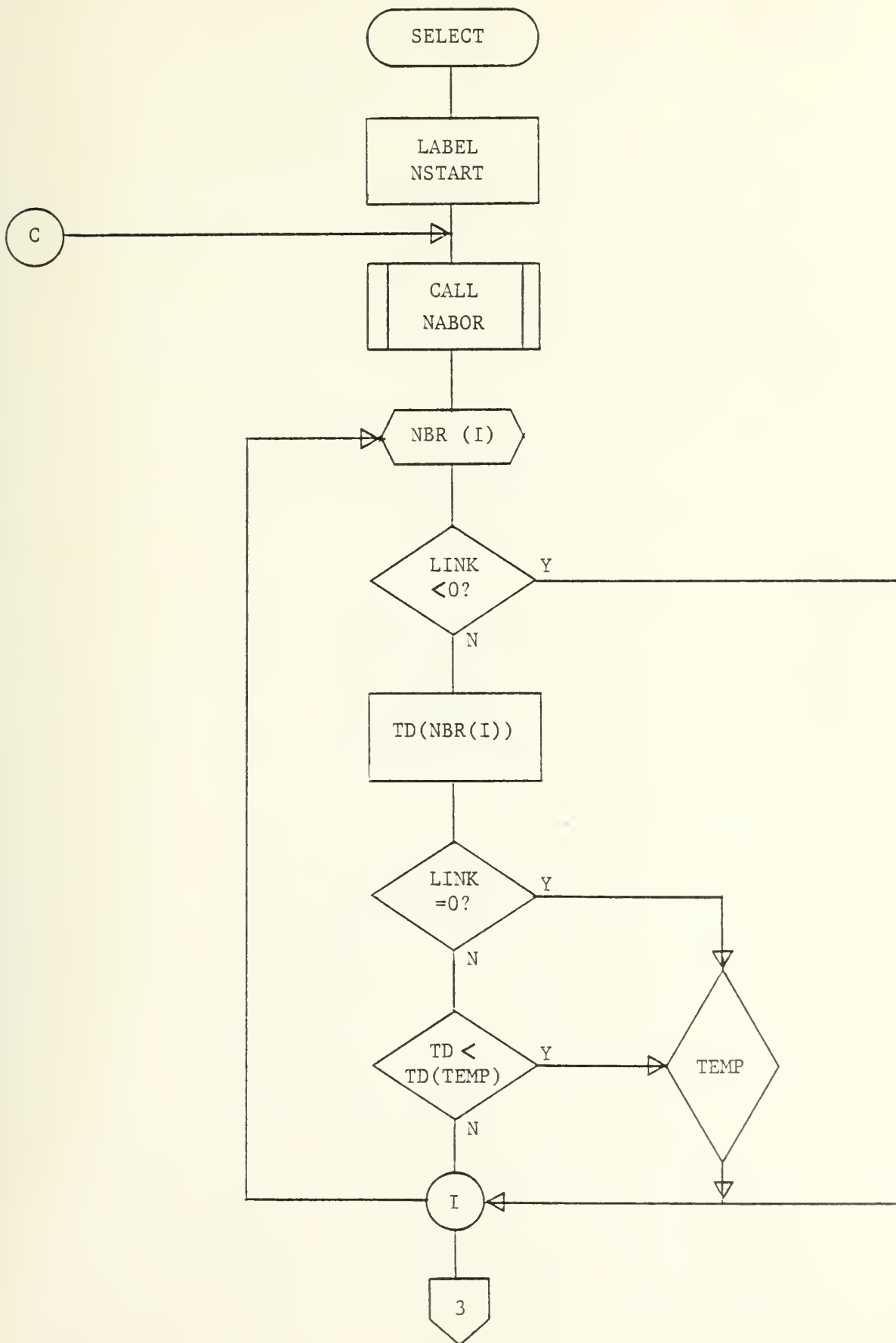




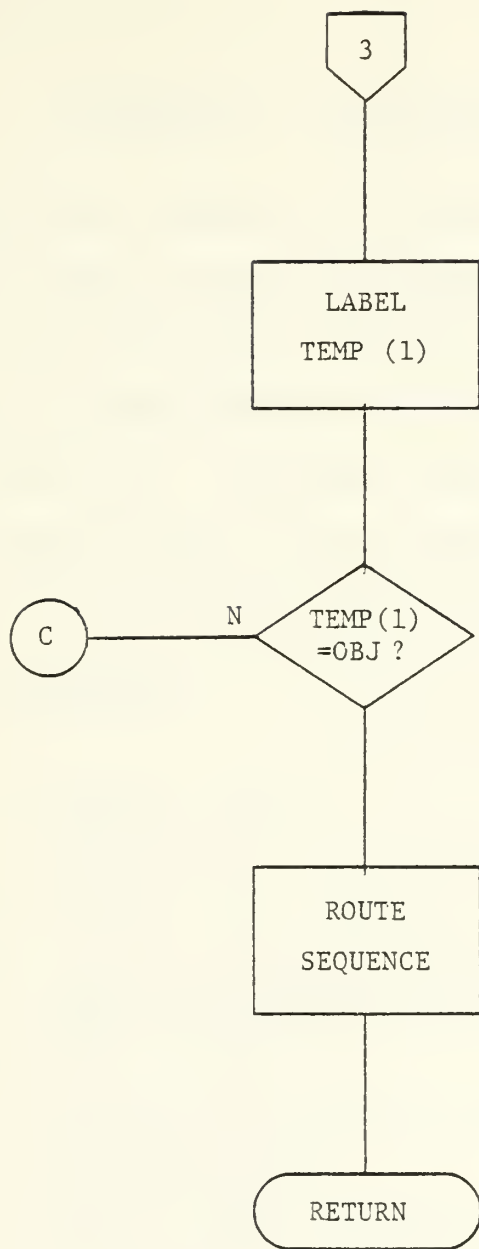












APPENDIX B: LIST OF VARIABLES

This appendix presents an alphabetical list which provides definitions of the major variables that are used in the route selection model. Variables which are used as counters or as dummy arguments in subroutines are not included on this list. Variables used to store intermediate results of computations are also omitted from the list. Scalar and array variables are listed separately.

1. SCALAR VARIABLES

ASLT	- assault range
CFIX	- actual column spacing in route selection grid
CSPACE	- desired column spacing in route selection grid
DIFF	- cumulative tactical difficulty for any optimum route section
DIST	- distance criterion for route re-evaluation
E	- exposure weight
JHALF	- center column of route selection grid
LAST	- location of last entry in array TEMP
MAX	- largest node number within distance criterion of NSTART
MIN	- smallest node number in current route selection area
NCALL	- type of route selection (initial or re-evaluation)
NCHG	- number of entries in array CHANGE
NCOL	- number of columns in route selection grid
NEXT	- location of first entry in array TEMP

NNBR - number of allowable route segments from any node
 NNODE - number of nodes in route selection grid
 NOBJ - node number of terminal point in route selection grid
 NPERM - node number of permanently labeled node
 NROW - number of rows in route selection grid
 NSTART - node number of maneuver element's current position
 NSTAT - number of entries in array STATUS
 NSUSP - number of entries in array SUSP
 NUM - number of nodes comprising an optimum route
 PCT - fraction of maneuver element visible to enemy influence
 RFIX - actual row spacing in route selection grid
 RNG - distance between any node and any source of enemy influence
 RSPACE - desired row spacing in route selection grid
 SECTOR - width of route selection grid
 T - travel time
 TD - temporary cumulative tactical difficulty for any route
 TYPE - type of enemy influence
 X,Y - coordinates of current position of maneuver element
 XO,YO - coordinates of initial position of maneuver element
 XT,YT - coordinates of terminal point in route selection grid

2. ARRAY VARIABLES

ANODE - attributes of nodes in route selection grid

Dimension
(300,15)

Col. 1 - X coordinate
2 - Y coordinate
3 - Z coordinate
4 - flag value
5 - E
6 thru 14 - travel times to allowable
neighbor nodes
15 - TD

ARC - relative locations of allowable neighbor nodes

(6,3)

Row 1 - relative column locations
2 - number of neighbors per column
3 thru 6 - relative row locations

CHANGE - attributes of temporary sources of enemy influence

(10,5)

Col. 1 - TYPE
2 - X coordinate
3 - Y coordinate
4 - Z coordinate
5 - flag value

NBR - neighboring nodes of any given node

(9)

LINK - predecessor nodes in shortest-route algorithm

(300)

NODES - sequence of node numbers comprising an optimum route

(50)

RTE - sequence of coordinates comprising an optimum route

(50,2)

Col. 1 - X coordinate
2 - Y coordinate

STATUS - input array of temporary enemy influences

(10,3)

Col. 1 - X coordinate
2 - Y coordinate
3 - Z coordinate



SUSP - attributes of suspected sources of influence
(10,2)

Col. 1 - X coordinate
2 - Y coordinate

TEMP - temporarily labeled nodes (300,3)

```
Col. 1 - node number
      2 - row location of node with next smaller TD
      3 - row location of node with next larger TD
```



APPENDIX C. COMPUTER PROGRAM

```

SUBROUTINE ROUTE (X,Y,XT,YT,NCALL,SECTOR,ASLT,RTE)
COMMON /EXT/ NSUSP,SUSP,NSTAT,STATUS
COMMON /INS/ ANODE,NNBR,NOBJ,MIN,NCOL,NNODE
DIMENSION RTE (50,2),ANODE (300,15),SUSP (10,3),
1 STATUS (10,3),NBR (9),NODES (50),CHANGE (10,5)
DATA DIST,RSPACE,CSPACE /750.0,100.0,100.0/
NNBR=9
IF (NCALL.EQ.1) GO TO 150

```

```

C
C IF INITIAL CALL, BEGIN INITIALIZATION PROCESS
C COMPUTE ARRAY PARAMETERS
C

```

```

XO=X
YO=Y
D=SQRT ((XT-X)**2+(YT-Y)**2)
NROW=INT (D/RSPACE+0.5) +1
RFIX=D/(NROW-1)
COL=SECTOR/CSPACE
NCOL=INT (COL+0.5) +1
IF (MOD (NCOL,2).NE.0) GO TO 20
IF (COL.LT.FLOAT (NCOL)) GO TO 10
NCOL=NCOL+1
GO TO 20
10 NCOL=NCOL-1
20 CFIX=SECTOR/(NCOL-1)
JHALF=(NCOL+1)/2
NNODE=NROW*NCOL
MIN=1

```



```

C
C  COMPUTE NODE COORDINATES
C
      N=1
      DO 30 I=1,NROW
      DO 30 J=1,NCOL
      ANODE(N,1) = X + ( (XT-X)*(I-1)/(NROW-1) ) +
1 ( (J-JHALF)*(YT-Y)*CFIX/(RFIX*(NROW-1)) )
      ANODE(N,2) = Y + ( (I-1)*(YT-Y)/(NROW-1) ) -
1 ( (J-JHALF)*(XT-X)*CFIX/(RFIX*(NROW-1)) )
      CALL ELEV(ANODE(N,1),ANODE(N,2),ANODE(N,3))
      ANODE(N,4)=0.0
      ANODE(N,5)=0.0
30    N=N+1
C
C  FLAG NODES WITHIN ASSAULT RANGE OF OBJECTIVE
C
      KROW=INT(ASLT/RFIX)+1
      IGO=NCOL*(NROW-KROW)+1
      IF(IGO.LT.1) IGO=1
      DO 40 I=IGO,NNODE
      D=SQRT( (ANODE(I,1)-XT)**2 + (ANODE(I,2)-YT)**2)
      IF(D.LE.ASLT) ANODE(I,4)=1.0
40    CONTINUE
C
C  COMPUTE EXPOSURE FACTORS & TRAVEL TIMES
C
      IF(NSUSP.EQ.0) GO TO 60
      DO 50 I=1,NSUSP
      CALL ELEV(SUSP(I,1),SUSP(I,2),SUSP(I,3))
50    CONTINUE
60    NCHG=NSTAT

```



```

DO 70 I=1,10
CHANGE(I,1)=0.0
IF(NSTAT.EQ.0) GO TO 70
IF(I.GT.NSTAT) GO TO 70
CHANGE(I,1)=STATUS(I,1)
CHANGE(I,2)=STATUS(I,2)
CHANGE(I,3)=STATUS(I,3)
CALL ELEV(CHANGE(I,2),CHANGE(I,3),CHANGE(I,4))
CHANGE(I,5)=999.0
70  CONTINUE
80  DO 140 I=1,NNODE
    IF(ANODE(I,4).EQ.1.0) GO TO 120
    IF(NSUSP.EQ.0) GO TO 100
    DO 90 J=1,NSUSP
        CALL EXPOSE(ANODE(I,1),ANODE(I,2),ANODE(I,3),1.0,
1  SUSP(J,1),SUSP(J,2),SUSP(J,3),E)
90    ANODE(I,5)=ANODE(I,5)+E
100   IF(NSTAT.EQ.0) GO TO 120
        DO 110 J=1,NCHG
            CALL EXPOSE(ANODE(I,1),ANODE(I,2),ANODE(I,3),
1  CHANGE(J,1),CHANGE(J,2),CHANGE(J,3),CHANGE(J,4),E)
110   ANODE(I,5)=ANODE(I,5)+E
120   CALL NABOR(I,MIN,NCOL,NNODE,NBR)
        DO 130 J=1,NNBR
            IF(NBR(J).LE.0) GO TO 130
            JJ=J+5
            CALL TIME(ANODE(I,1),ANODE(I,2),ANODE(I,3),
1  ANODE(NBR(J),1),ANODE(NBR(J),2),D,ANODE(I,JJ))
130   CONTINUE
140   CONTINUE
C
C  SELECT INITIAL ROUTE
C
    NSTART=JHALF
    NOBJ=NNODE-JHALF+1
    NCALL=1

```



```

      GO TO 240

C
C  IF SUBSEQUENT CALL, BEGIN RE-EVALUATION PROCESS
C  COMPUTE RE-EVALUATION PARAMETERS
C
150   I=INT(1.5+(((XT-XO)*(X-XO)+(YT-YO)*(Y-YO))/
1     (RFX**2*(NROW-1))))
      J=INT(JHALF+0.5+(((YT-YO)*(X-XO)-(XT-XO)*(Y-YO))/
1     (RFX*CFIX*(NROW-1))))
      NSTART=((I-1)*NCOL)+J
      MIN=((I-2)*NCOL)+1
      ANODE(NSTART,1)=X
      ANODE(NSTART,2)=Y
      CALL ELEV(X,Y,ANODE(NSTART,3))
      CALL NABOR(NSTART,MIN,NCOL,NNODE,NBR)
      DO 160 K=1,NNBR
      IF(NBR(K).LE.0) GO TO 160
      CALL TIME(X,Y,ANODE(NSTART,3),ANODE(NBR(J),1),
1 ANODE(NBR(J),2),D,T)
      KK=K+5
      ANODE(NBR(J),KK)=T
160   CONTINUE

C
C  UPDATE TACTICAL CHANGES
C
      IF(NSTAT.EQ.0) GO TO 205
      DO 200 I=1,NSTAT
      DO 170 J=1,10
      IF(CHANGE(J,1).EQ.0.0) GO TO 170
      IF(STATUS(I,1).NE.CHANGE(J,1)) GO TO 170
      IF(STATUS(I,2).NE.CHANGE(J,2)) GO TO 170
      IF(STATUS(I,3).NE.CHANGE(J,3)) GO TO 170
      CHANGE(J,5)=0.0
      GO TO 200
170   CONTINUE

```



```

DO 180 K=1,10
IF (CHANGE (K,1) .EQ.0.0) GO TO 190
180 CONTINUE
190 CHANGE (K,1)=STATUS (I,1)
CHANGE (K,2)=STATUS (I,2)
CHANGE (K,3)=STATUS (I,3)
CALL ELEV (CHANGE (K,2),CHANGE (K,3),CHANGE (K,4))
CHANGE (K,5)=1.0
IF (K.GT.NCHG) NCHG=K
200 CONTINUE
205 DO 210 I=1,10
IF (CHANGE (I,1) .EQ.0.0) GO TO 210
IF (CHANGE (I,5) .EQ.999.0) CHANGE (I,5)=-1.0
210 CONTINUE
C
C UPDATE EXPOSURE FACTORS
C
DO 225 I=MIN,NNODE
IF (ANODE (I,4) .EQ.1.0) GO TO 225
DO 220 J=1,10
IF (CHANGE (J,1) .EQ.0.0) GO TO 220
IF (CHANGE (J,5) .EQ.0.0) GO TO 220
CALL EXPOSE (ANODE (I,1),ANODE (I,2),ANODE (I,3),
1 CHANGE (J,1),CHANGE (J,2),CHANGE (J,3),CHANGE (J,4),E)
ANODE (I,5)=ANODE (I,5)+E*CHANGE (J,5)
220 CONTINUE
225 CONTINUE
DO 230 I=1,NCHG
IF (CHANGE (I,5) .EQ.- 1.0) CHANGE (I,1)=0.0
230 CHANGE (I,5)=999.0
C
C SELECT OPTIMUM ROUTE
C
240 CALL SELECT (NSTART,NUM,NODES)

```



```

C
C   RECORD ROUTE COORDINATES
C
      IROW=INT(DIST/RPIX)+1
      MAX=MIN+IROW*NCOL
      IF (MAX.LT.NNODE) GO TO 250
      J=NUM
      GO TO 270
250   DO 260 J=1,NUM
      IF (NODES(J).LE.MAX) GO TO 270
260   CONTINUE
270   DO 280 I=1,J
      K=NUM-I+1
      RTE (I,1)=ANODE (NODES (K) ,1)
280   RTE (I,2)=ANODE (NODES (K) ,2)
      RETURN
      END

```



```

SUBROUTINE SELECT(NSTART,NUM,NODES)
COMMON /INS/ ANODE,NNBR,NOBJ,MIN,NCOL,NNODE
DIMENSION ANODE(300,15),NODES(50),LINK(300)
1 TEMP(300,3),NBR(9)
DO 5 I=1,300
LINK(I)=0
ANODE(I,15)=9999.0
5 TEMP(I,1)=0.0
NPERM=NSTART
NEXT=0
LAST=0
DIFF=0.0
LINK(NPERM)=-999
C
C RECORD NEXT SET OF TEMPORARY LABELS
C
10 CALL NABOR(NPERM,MIN,NCOL,NNODE,NBR)
DO 100 I=1,NNBR
IF(NBR(I).EQ.-1) GO TO 100
IF(LINK(NBR(I)).LT.0) GO TO 100
II=I+5
TD=ANODE(NPERM,II)*(1.0+ANODE(NBR(I),5))+DIFF
IF(ANODE(NBR(I),15).LE.TD) GO TO 100
ANODE(NBR(I),15)=TD
IF(LINK(NBR(I)).EQ.0) GO TO 20
DO 15 J=1,300
IF(TEMP(J,1).NE.FLOAT(NBR(I))) GO TO 15
IF(J.EQ.NEXT) GO TO 90
TEMP(INT(TEMP(J,2)),3)=TEMP(J,3)
IF(J.EQ.LAST) GO TO 13
TEMP(INT(TEMP(J,3)),2)=TEMP(J,2)
GO TO 45
13 LAST=INT(TEMP(J,2))
GO TO 45
15 CONTINUE
20 DO 30 J=1,300

```



```

        IF (TEMP (J, 1) .EQ. 0.0) GO TO 40
30      CONTINUE
40      TEMP (J, 1) = FLOAT (NBR (I) )
45      IF (NEXT.EQ.0) GO TO 70
        IF (TD.LE.ANODE (INT (TEMP (NEXT, 1) ) , 15) ) GO TO 70
        IF (TD.GE.ANODE (INT (TEMP (LAST, 1) ) , 15) ) GO TO 80
        K=INT (TEMP (NEXT, 3) )
50      IF (TD.GT.ANODE (INT (TEMP (K, 1) ) , 15) ) GO TO 60
        TEMP (J, 3) =FLOAT (K)
        TEMP (J, 2) =TEMP (K, 2)
        TEMP (INT (TEMP (K, 2) ) , 3) =J
        TEMP (K, 2) =FLOAT (J)
        GO TO 90
60      K=INT (TEMP (K, 3) )
        GO TO 50
70      TEMP (J, 2) =0.0
        TEMP (J, 3) =FLOAT (NEXT)
        IF (NEXT.NE.0) GO TO 75
        NEXT=J
        LAST=J
        GO TO 90
75      TEMP (NEXT, 2) =FLOAT (J)
        NEXT=J
        GO TO 90
80      TEMP (J, 3) =0.0
        TEMP (J, 2) =FLOAT (LAST)
        TEMP (LAST, 3) =FLOAT (J)
        LAST=J
90      LINK (NBR (I) ) =NPERM
100     CONTINUE
C
C      RECORD NEXT PERMANENT LABEL
C
        NPERM=INT (TEMP (NEXT, 1) )
        IF (NPERM.EQ.NOBJ) GO TO 110
        TEMP (NEXT, 1) =0.0

```



```

NN=INT (TEMP (NEXT, 3) )
TEMP (NEXT, 3) =0.0
NEXT=NN
TEMP (NEXT, 2) =0.0
DIFF=ANODE (NPERM, 15)
LINK (NPERM) =-LINK (NPERM)
GO TO 10

```

C

C RECORD OPTIMUM ROUTE SEQUENCE

C

```

110 DO 120 I=1,100
    NODES (I) =NPERM
    NPERM=IABS (LINK (NPERM) )
    IF (NPERM.EQ.999) GO TO 130
120 CONTINUE
130 NUM=I
    RETURN
    END

```



```

SUBROUTINE NABOR(NODE,MIN,NCOL,NNODE,NBR)
DIMENSION NBR(1),ARC(6,3)
DATA ARC /-1.,4.,-1.,0.,1.,2.,0.,1.,1.,3*0.,1.,4.,-1.,
1 0.,1.,2./
I=1
J=1
IF(NODE.GT.1) GO TO 20
K=ARC(2,1)
DO 10 L=1,K
NBR(I)=-1
10 I=I+1
J=2
20 M=MOD(NODE,NCOL)
DO 50 N=J,3
II=NODE+ARC(1,N)
JJ=ARC(2,N)
DO 40 KK=1,JJ
IF(M.EQ.1 .AND. N.EQ.1) GO TO 30
IF(M.EQ.0 .AND. N.EQ.3) GO TO 30
LL=KK+2
NBR(I)=II+NCOL*ARC(LL,N)
IF(NBR(I).GT.NNODE .OR. NBR(I).LT.MIN) NBR(I)=-1
GO TO 40
30 NBR(I)=-1
40 I=I+1
50 CONTINUE
RETURN
END

```



```

SUBROUTINE EXPOSE(X1,Y1,Z1,TYPE,X2,Y2,Z2,E)
DIMENSION WEIGHT(4,2)
DATA WEIGHT /3.,6.,4.,2.,3000.,2500.,3000.,1500./
E=0.0
CALL LOS(X1,Y1,Z1,X2,Y2,Z2,PCT)
IF(PCT.GT.0.1) E=1.0
NTYPE=INT(TYPE)
RNG=SQRT((X1-X2)**2+(Y1-Y2)**2)
IF(RNG.GT.WEIGHT(NTYPE,2)) RETURN
E=E+WEIGHT(NTYPE,1)-RNG*WEIGHT(NTYPE,1)/
1 WEIGHT(NTYPE,2)
IF(NTYPE.NE.3) RETURN
IF(RNG.GT.WEIGHT(4,2)) RETURN
E=E+2.0-RNG*2.0/1500.0
RETURN
END

```


LIST OF REFERENCES

1. Department of the Army, Review of Selected Army Models, by J. Honig, and others, May 1971.
2. Systems Research Group Report AR 69-2A, The Tank Weapon System Final Report, by A. B. Bishop and G. M. Clark, 1969.
3. Dreyfus, S. E. and Law, A. M., The Art and Theory of Dynamic Programming, Academic Press, 1977.
4. Nemhauser, G. L., Dynamic Programming, Wiley, 1966.
5. Dreyfus, S. E., "An Appraisal of Some Shortest-Path Algorithms", Operations Research, v.17, p.395-417, May 1969.
6. Golden, B., "Shortest-Path Algorithms: A Comparison", Operations Research, v. 24, p. 1164-1168, November 1976.
7. Pape, U., "Implementation and Efficiency of Moore-Algorithms for the Shortest Route Problem", Mathematical Programming, v. 7, p. 212-222, October 1974.



INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
4. Professor S. H. Parry, Code 55PY (thesis advisor) Department of Operations Research Naval Postgraduate School Monterey, California 93940	5
5. Professor J. K. Hartman, Code 55HH (thesis advisor) Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
6. CPT J. S. Kramer 905 W. 18 Sioux Falls, South Dakota 57105	1
7. LTC E. P. Kelleher, Code 55KA Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
8. Office of the Commanding General U. S. Army TRADOC Attn: General Donn A. Starry Ft. Monroe, Virginia 23651	1
9. Headquarters U. S. Army Training & Doctrine Command Attn: ATCG-T (Col. Ed Scribner) Ft. Monroe, Virginia 23651	1

10. Headquarters 1
U. S. Army Training & Doctrine Command
Attn: Director, Analysis Directorate -
Combat Developments (Col. Tony Pokorney)
Ft. Monroe, Virginia 23651
11. Headquarters 1
U. S. Army Training & Doctrine Command
Attn: Director, Maneuver Directorate -
Combat Developments (Col. Fred Franks)
Ft. Monroe, Virginia 23651
12. Lt. General J. R. Thurman 1
Commanding General
U. S. Army Combined Arms Center
Ft. Leavenworth, Kansas 66027
13. Director 1
Combined Arms Combat Development Activity
Attn: Col. Reed
Ft. Leavenworth, Kansas 66027
14. Dr. Wilbur Payne, Director 1
U. S. Army TRADOC Systems Analysis Activity
White Sands Missile Range, New Mexico 88002
15. Director 1
Armored Combat Vehicle Technology Program
Attn: Col. Fitzmorris
U. S. Army Armor Center
Ft. Knox, Kentucky 40121
16. Director 1
Studies Division - Combat Developments
Attn: Col. Burnett
U. S. Army Armor Center
Ft. Knox, Kentucky 40121
17. Col. Frank Day 1
TRADOC Systems Manager - XML
U. S. Army Armor Center
Ft. Knox, Kentucky 40121
18. MG Dick Lawrence 1
Tank Forces Management Office
Room 1A-871, The Pentagon
Washington, D.C. 20310
19. Mr. David Hardison 1
Deputy Undersecretary of the Army
(Operations Research)
Department of the Army, The Pentagon
Washington, D.C. 20310



20. Director 1
U. S. Army Material Systems Analysis Activity
Attn: Mr. Will Brooks
Aberdeen Proving Grounds, Maryland 21005
21. Command and General Staff College 1
Attn: Education Advisor
Room 123, Bell Hall
Ft. Leavenworth, Kansas 66027
22. Headquarters, Department of the Army 1
Office of the Deputy Chief of Staff
for Operations and Plans
Attn: DAMO-2D
Washington, D.C. 20310





Thesis
K8475
c.1

Kramer

280831

Simulation of
dynamic tactical
route selection with
application in the
STAR model.

